IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 24, 2020, accepted May 4, 2020, date of publication May 11, 2020, date of current version May 26, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2993688

How to Hide the Real Receiver Under the Cover Receiver: CP-ABE With Policy Deniability

PO-WEN CHI[1], MING-HUNG WANG[2], AND HUNG-JR SHIU[3]
[1]Department of Computer Science and Information Engineering, National Taiwan Normal University, Taipei 11677, Taiwan
[2]Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan
[3]Department of Computer Science, Tunghai University, Taichung 40704, Taiwan

Corresponding author: Hung-Jr Shiu (hjshiu@thu.edu.tw)

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 107-2218-E-003-002-MY3 and Grant MOST 107-2218-E-035-009-MY3, and in part by the Taiwan Information Security Center, National Sun Yat-sen University (TWISC@NSYSU).

ABSTRACT Attribute-based encryption (ABE) is a useful tool for sharing an encrypted data to a target group. In a ciphertext-policy ABE (CP-ABE) scheme, a ciphertext includes a policy to indicate its receivers and only those receivers can correctly decrypt the ciphertext. Since this design leaks the receiver identity, it may raise a new security issue about user privacy. Some hidden-policy ABE schemes, where the policy is secretly protected, are proposed to keep user privacy. However, these hidden-policy ABE schemes rely on the user trying all possibilities to decide if it belongs to the wanted receiver group. The decryption costs too much and every potential receiver will run the decryption process in vain since it does not know the policy. In this work, we apply the deniability concept to solve this problem. The encryption scheme allows the sender to claim the ciphertext is for some receiver group while actually it is for another receiver group. Both receiver groups can correctly decrypt the ciphertext except that the real group can get the real message and the cover group will get the cover message. While coercion, the sender can definitely claim the ciphertext is for the cover group and the real group is kept confidential.

INDEX TERMS Attribute-based encryption, deniable encryption, identity-based encryption.

I. INTRODUCTION

Encryption techniques are useful tools to protect data confidentiality. Generally speaking, a sender and a receiver need to share information before they can communicate securely. For example, in a symmetric-encryption key scheme, the sender and the receiver share the same key. For an asymmetric-key encryption scheme, both the sender and the receiver share a public system parameter and the receiver's public key where the public key delivery may be through a public key infrastructure. Therefore, once a ciphertext is generated, the ciphertext is committed to a static message and a receiver. If the ciphertext is intercepted by the authority, even though the authority knows nothing about the key, it may have the power to force both the sender and the receiver to *open* the ciphertext.

To solve this problem, non-committing encryption [1] and deniable encryption [2] have been proposed. These two encryptions enable users to open an existing ciphertext belonging to the particular fake message. That is, the ciphertext will not be committed to the actual message. Though this kind of solution solves the message secrecy under coercion issue, the receiver identity is public in the ciphertext and may be possible to leak some important information. Take a CP-ABE scheme as an example. The ciphertext of a CP-ABE scheme includes an explicit access policy, so an attacker may infer from the policy to get some receiver's sensitive attributes even if the ciphertext cannot be successfully decrypted. For example, in a personal health record protected by a CP-ABE scheme, the access policy may reveal sensitive information like dentist, clinic and so on.

Some hidden CP-ABE (HCP-ABE) schemes were proposed to solve this problem [3]–[7]. These schemes make the attributes in the access policy unknown so there will be no information leakage. The trade-off is that since no one can learn the target receiver from the ciphertext directly, the user needs to run the decryption process to check if it belongs to the target receiver group. So the overall system cost increases greatly. In this paper, we develop another approach which is called **deniable policy**. The concept is motivated from

The associate editor coordinating the review of this manuscript and approving it for publication was Mostafa M. Fouda.

steganography [8]. That is, we do not want to make the access policy unknown. Instead, we want to make a fake policy to cover the real policy.

Steganography is a technique to conceal a file, message, image, or video within another file, message, image, or video, and it is also called data hiding. In steganography, two kinds of media are integrated to conceal data; the one contains a hidden message is called the **cover media** while the hidden message is called the **stego media**. Outsiders can only see the cover media and are unaware of the existence of the hidden media. This technique is often applied to digital right management services because copyright owners want to embed their signatures into their works without being noticed. Following this concept, in this paper, we ask a similar question, **is it possible to hide the real receiver identity under another cover receiver identity in one ciphertext?** That is, we wonder that is it possible to create a ciphertext that looks like a ciphertext for some claimed receivers but is actually for another receiver.

Fig.1 presents an example of steganography. Suppose Alice wants to secretly share a file with her friend Charles. If she simply encrypts the file and sends it to Charles, her mother, who we use a police icon in Fig.1 to represent, may ask her to decrypt the file. Moreover, Alice's mother may call Charles and ask him to reveal the content. To avoid this case, Alice can apply steganography for encrypting the file. Alice will first prepare two messages. One is an unimportant message for Bob, and the other is an actual secret message for Charles. Alice encrypts these two files and embeds Charles's ciphertext into Bob's ciphertext. Alice puts the processed ciphertext on a public channel and asks all her friends to download the file. Only Bob and Charles can successfully decrypt the ciphertext, but they derive different messages, one is the cover message and the other is the real message. The "successful decryption" means that both Bob and Charles can get meaningful messages, which are prepared by the sender, after their decryption operations. When questioned by her mother, Alice can claim the ciphertext is for Bob and reveals the message sent to Bob. Bob can also be an **honest** witness because he only knows what he received.

Even if Alice's mother believes that there is something hidden in the ciphertext, she cannot confirm which friend of Alice is the real receiver. Note that in this scenario, Alice does not need to initially collude with Charles since her mother cannot suspect Charles unless she suspects everyone.

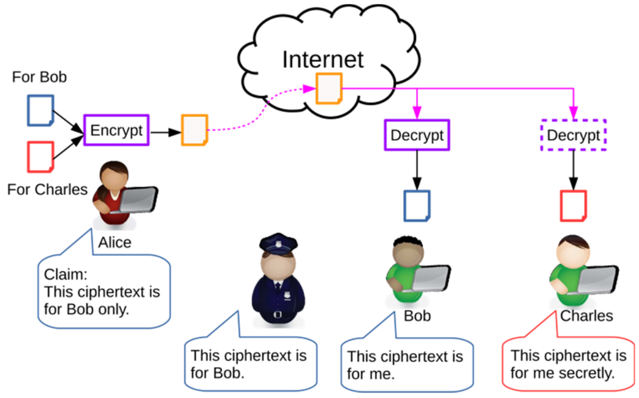
Internet
For Bob
Encrypt
For Charles
Decrypt
Decrypt
Alice
Claim:
This ciphertext is for Bob only.
Bob
Charles
This ciphertext is for Bob.
This ciphertext is for me.
This ciphertext is for me secretly.

FIGURE 1. CP-ABE with policy deniability scenario.

In this paper, we develop a ciphertext-deniable-policy ABE scheme (CDP-ABE) scheme which we do not call it deniable CP-ABE scheme since this name has been used only for data deniability. In a CDP-ABE scheme, a user prepares two messages for two access policies respectively. For simplicity, in this paper, we use the **real policy** and the **real message** to represent the policy and the message which should keep secret to outside coercers. We use the **cover policy** and the **cover message** to represent the pair that can be opened to outside coercers. Note that there is no receiver-deniability issues since CDP-ABE conceals the real communication targets.

Our CDP-ABE design applies the concept of multi-distributional deniable encryption. Our CDP-ABE scheme is composed of two sets of algorithms, including a normal encryption scheme and a deniable encryption scheme. The normal-set encryption scheme encrypts one message for one access policy, while the deniable-set encryption scheme encrypts two messages for two different policies. The output ciphertexts from both sets of algorithms are computationally indistinguishable. Therefore, the sender can claim that the ciphertext comes from the normal set and is for a particular receiver group while actually the ciphertext is from the deniable set which implies that there is actually another receiver group which can get different message from this ciphertext. The outsider is not able to challenge the sender's claims since the ciphertexts from two algorithm sets are indistinguishable. Moreover, the receiver for the cover policy can also be an innocent witness since it can successfully decrypt the ciphertext.

In this paper, we construct a CDP-ABE scheme to protect the access policy. Our contributions are listed as follows.

1) **Policy Deniability**. To the best of our knowledge, we are the first group to consider the issue of policy deniability, which is an important feature in deniable ABE encryption. The proposed scheme has at least two advantages over previous deniable encryption schemes. First, because the access policy is covered, there is no receiver-deniability issue since the outside coercer will be misled to other receiver group. So, the sender and the receiver do not need to concern about agreement issues. Especially when there are lots of potential receivers, agreement with every member takes lots of communication works. The second benefit is that the scheme creates an innocent receiver group, who is defined by the cover policy. The member in the cover receiver group believes that the ciphertext is dedicated for himself/herself and he/she can be the role as a witness to convince an outside coercer. Of course, there are some works about anonymous broadcast encryption as described in section II, and they have the similar benefit. However, those schemes do not consider the coercion issue.

2) **Different Messages Support**. Our CDP-ABE supports two different groups, where one is defined by the real policy and the other is defined by the cover policy. Undoubtedly, these two groups should derive two different messages from the same ciphertext or the real message is released to the outsider. Therefore, our CDP-ABE scheme forges not only the receiver policy but also the message. When being coerced, the sender can claim an existing ciphertext is a message for some group while actually the ciphertext is another message for another group.
3) **Fake Key Consistency**. Though there is no receiver-deniability issue in our scheme, the powerful coercer may force all service users to release their own keys. In our design, the released key is valid for all ciphertexts with the claimed access policies in these ciphertexts. Our scheme supports one encryption environment for many time uses. This is true for a normal key since the key can be used to decrypt all ciphertexts which are for the key owner. In our scheme, we ensure that the fake key has the similar property. That is, when a deniable service user releases its fake key, the key can be used to decrypt all ciphertexts which are claimed for the user, including normally encryption ciphertexts and deniably encryption ciphertexts. As for the ciphertexts which are intended for the user but claimed for others, the released user-fake key looks irrelevant to these ciphertexts.
4) **Different Deniable-Policy Size Support**. In a CDP-ABE scheme, there are two different policies, the real policy and the cover policy. These two policies may have different sizes. For example, the real policy may require three attributes while the cover policy may require five attributes. The attribute number can be treated as a kind of policy size. Undoubtedly, the sender can try to forge a cover policy which has the same size with the real policy. However, the forged policy looks strange because it is composed of some attributes that seldom appear together for matching the real policy size. In this paper, we introduce the redundant attribute idea and make the sender arbitrary modify the policy size to solve this problem.

The rest of our paper is organized as follows. In Section II, we review some related studies including deniable encryption, deniable authentication, broadcast encryption and HCP-ABE. In Section III, we propose formal definitions of CDP-ABE and the properties that it must satisfy. Then, we construct our scheme in Section V with correctness verification, security and deniability proof, and performance evaluation. We provide a generalization discussion in Section VI, and the last section is our conclusion.

II. RELATED WORKS

In this section, we review related works including deniable encryption, deniable authentication, broadcast encryption, deniable ABE and HCP-ABE.

A. DENIABLE ENCRYPTION

The idea of deniable encryption was first proposed by Canetti *et al.* [2]. Deniable encryption can be divided into a deniable shared key scheme and a public key scheme. The one-time pad is a simple example of deniable encryption. Let a message m be encrypted into a ciphertext c via $c = m \oplus k$, where k is the shared key. The encryptor can claim that the message is m' with the key $k' = c \oplus m'$ and no one can challenge this claim. For the deniable-public-key encryption scheme, the sender and/or the receiver need to provide evidence for their claims. In this paper, we review several important deniable-public-key encryption schemes.

In the scheme proposed by Canetti *et al.* [2], the authors used a translucent set to provide fake messages with convincing evidence. A translucent set is a set that contains a trapdoor subset. It is easy to pick a random element from the universal set or the subset; however, it is hard to determine if an element belongs to the subset without the trapdoor. If a sender wants to encrypt one bit, the sender sends an element not contained in the subset. To encrypt one bit 1, the sender sends an element contained in the subset. When coerced, the sender claims a bit from 1 to by claiming the random element from the universal set that coincidentally lies in the subset. Canetti *et al.* called this scheme *sender-deniable* which means that the scheme allows the sender to provide evidence for fake messages. Canetti *et al.* also extended the scheme through an interactive approach to support *receiver-deniability* and combined them into a *bi-deniable* encryption scheme. Following this idea, many researchers have used different tools to build translucent sets. Dürmuth and Freeman [9] used samplable encryption to construct a translucent set. O'Neill *et al.* [10] developed a bi-translucent set based on a lattice. Klonowski *et al.* improved the scheme of Canetti *et al.* to support messages at any depth [11].

In addition to translucent set approaches, there are other proposed techniques to build deniable encryption schemes. O'Neill *et al.* [10] made use of a simulatable public-key system and a voting approach to provide deniability. The simulatable public-key system provides an oblivious key generation function and an oblivious ciphertext function. Gasti *et al.* [12] proposed another deniable scheme where the system claims to set up one public-private key pair while there are two pairs. The sender decides which key is released according to the outside coercer. Ibrahim used the quadratic residuosity problem to provide deniability [13]. Chi and Lei [14] proposed a decryption scheme based on composite order groups. In this scheme, the real data and pre-determined fake data are hidden in the different subgroups in one composite order group.

In a recent paper [15], Canetti *et al.* proposed a new deniable feature called *off-the-record deniability*. This work is the first approach that allows the sender and the receiver to make different claims. With this method, an outside coercer cannot determine who is lying.

However, the objective of the CDP-ABE encryption scheme is somewhat different from that of the deniable encryption scheme. The CDP-ABE encryption scheme is designed to keep the message secret under coercion just like the deniable encryption scheme; however, the CDP-ABE encryption scheme also protects the receiver's identity by hiding the real policy behind a cover policy. In this way, there is no need for receiver-deniability or off-the-record deniability.

B. DENIABLE AUTHENTICATION

Even though encryption and authentication are often based on similar techniques, deniable authentication is entirely different from deniable encryption. Deniable authentication is a technique that allows the sender and the receiver to authenticate each other; however, they cannot convince a third party with the authentication process. This concept was first proposed by Dwork *et al.* [16]. The basic idea is *zero-knowledge proof.*

Many deniable authentication schemes have been presented. Noar presented deniable ring authentication with ring signatures [17]. Deng *et al.* proposed two deniable authentication schemes based on the factoring problem and the discrete logarithm problem [18]. Fan *et al.* built a deniable authentication scheme using the Diffie-Hellman key exchange protocol [19]. Following a similar idea, Shao developed a deniable authentication protocol based on a generalized ElGamal signature scheme [20]. Xiao *et al.* used a chaotic encryption-hash parallel algorithm and the semi-group property of the Chebyshev chaotic map to design a deniable authentication scheme [21]. Raimondo *et al.* focused on IPSec key-exchange protocol and implemented a deniable authentication feature within it [22]. Raimondo *et al.* also proposed a new feature called *forward deniability* [23].

When deniable authentication schemes focus on user authentication deniability, they aim to deny communication entities with the ability to prove the target identity to third parties. However, it is easy for an outsider to be aware of the communication. Also, even if the exchange of a message reveals nothing, outsiders may trace the network traffic to obtain the user's identity. Our CDP-ABE encryption allows the sender to set up a strawman to deceive the adversary. Therefore, an outside coercer will not perceive the existence of the real receiver. Moreover, the sender does not need to agree with the cover receiver; in this way, the cover receiver is completely innocent. That is, compared to deniable authentication, we provide a fake receiver to third parties.

C. BROADCAST ENCRYPTION AND MULTI-RECEIVER ENCRYPTION

Broadcast Encryption is an encryption technique that confidentially delivers a message to a subset from a universe of users. The concept was first proposed by Fiat and Naor [24]. ABE can also be treated as one kind of broadcast encryption. Another similar encryption scheme is called multi-receiver encryption which allows multiple receivers to derive the message from one ciphertext [25], [26].

Generally speaking, the receiver set is embedded in the broadcast encryption ciphertext. That is, even non-target users can recognize the receiver group of a ciphertext. So, how to keep the receiver identity secret is a big challenge. There are lots of works about receiver anonymity encryption schemes, such as [27]–[30].

However, anonymous broadcast encryption, including anonymous multi-receiver encryption, focuses only on receiver identity hiding. The message in one ciphertext is the same for every receiver. So if one of the target receivers is compromised, the message is still released even the other receiver identities are kept unknown. Besides, they do not care about the coercion issue. As for our proposed work, CDP-ABE tries to use a valid ciphertext to cover both **different receivers** and **different messages**. So our work can be treated as a ciphertext hiding technique, where one ciphertext is hidden in another cover ciphertext. According to the previous studies, to the best of our knowledge, our proposed encryption system has not been presented before.

D. ATTRIBUTE-BASED ENCRYPTION

Attribute based encryption (ABE) is widely used to protect the security of personal data. Authors proposed an ABE scheme with full verifiability for outsourced decryption, which can simultaneously check the correctness for transformed ciphertext belongs to the authorized users and unauthorized users [31]. The proposed ABE scheme with verifiable outsourced decryption is proved to be selective CPA-secure in the standard model.

Authors gave the formal definition and security model of hierarchical attribute-based encryption (HABE) with continuous leakage-resilience in [32]. They presented a ciphertext-policy HABE scheme with continuous leakage-resilience. The scheme is resilient to master key leakage and secret key leakage. They also proved the security of the scheme under composite-order bilinear group assumptions using dual-system encryption techniques. The performance of leakage-resilience is analyzed theoretically.

Authors provide a ciphertext-policy attribute-based encryption (CP-ABE) scheme with efficient user revocation for cloud storage system [33]. User revocation is solved efficiently by introducing the concept of user group. The scheme is designed to outsource high computation load to cloud service providers without leaking file content and secret keys. Also, the scheme can withstand collusion attack performed by revoked users cooperating with existing users. The security of the proposed scheme under the divisible computation Diffie-Hellman assumption is proved and computation cost for local devices is relatively low.

Authors presented a user collusion avoidance ciphertext-policy ABE scheme with efficient attribute revocation for the cloud storage system [34]. The problem of attribute revocation is solved efficiently by exploiting the concept of an attribute group. The proposed scheme is proved secure against collusion attack launched by the existing users and the revoked users. The security of the proposed scheme is also reduced to the computational Diffie-Hellman assumption.

Authors proposed a ciphertext-policy attribute-based encryption (CP-ABE) scheme that enables fine-grained access control of encrypted IoT data on cloud [35]. They first presented an access control system model of CloudIoT platform by using ABE. Based on the proposed system model, the authors constructed a ciphertext-policy hiding CP-ABE scheme, which guarantees the privacy of the users. They further constructed a white-box traceable CP-ABE scheme with accountability in order to address the user key abuse and authorization center key abuse. The proposed systems are efficient according to their experiments.

Authors construct a flexibly bi-deniable Attribute-Based Encryption (ABE) scheme for all polynomial-size branching programs from learning with errors (LWE) [36]. The techniques involve new ways of manipulating Gaussian noise that may be of independent interest, and lead to a significantly sharper analysis of noise growth in Dual Regev type encryption schemes.

Authors presented a design for a new cloud-storage encryption scheme that enables cloud storage providers to create convincing-fake user secrets to protect user privacy [37]. Since coercers cannot tell if obtained secrets are true or not, the cloud storage provider ensure that user privacy is still securely protected.

E. HIDDEN CIPHERTEXT POLICY ATTRIBUTE-BASED ENCRYPTION

In most CP-ABE schemes, the policy is directly appended in the ciphertext. Though the message of the ciphertext is secret, sometimes the policy reveals the information of the message. To solve this problem, some HCP-ABE schemes are proposed.

The first CP-ABE with hidden access policy was proposed by Nishide *et al.* [3]. They made the access policy not embedded in the ciphertext. Instead, they listed all attributes in a randomized form. So the attacker needs to try all possibilities to reconstruct the policy. Another technique is to use the composite order group. The access policy is encoded in one subgroup and is randomized with other subgroup elements [4]–[6]. Since the access policy is hidden, the decryptor needs to do lots of computation to see if he belongs to the receiver group or not. To speed up the decryption process, Zhang *et al.* refined the access structure with separating the attribute name and the attribute value [7] so the decryptor only needs to check the attribute key according to the attribute name.

Though both CDP-ABE and HCP-ABE want to protect the access policy, the protection approaches are totally different. HCP-ABE tries to hide the access policy in some randomized form while we try to hide the access policy behind the cover policy. In our approach, a user does not need to doubt if it is the receiver or not before running the decryption algorithm. So our approach is definitely more practical than HCP-ABE since users in HCP-ABE will do lots of meaningless decryption works.

III. DEFINITIONS

In this section, we define the CDP-ABE scheme and its security model.

A. CDP-ABE DIFINITION

The objective of CDP-ABE is to conceal the real receiver from outsiders. To hide the receiver's identity and the real message, CDP-ABE allows the sender to create a valid ciphertext for another receiver group with a pre-determined fake message. The policy that defines the fake receiver group is called the cover policy and the ciphertext as the *carrier ciphertext*. Next, the sender embeds the real ciphertext, that is, the ciphertext for the real receiver group, in the carrier ciphertext. The sender can publish this processed ciphertext to a public channel and claim that this ciphertext is for the cover receiver group. The real receiver can obtain the ciphertext from the public channel and acquire the actual message. The cover receiver recovers only the pre-determined fake message.

Unlike other deniable encryption schemes, receiver-deniability is not an issue with CDP-ABE, because the real receiver is obscured and no one but the sender knows the receiver's identity. Even if an outside coercer compromises the cover receiver, they cannot determine anything about the real receiver from the "fake" ciphertext and therefore it is impossible to identify the receiver. This idea originates from steganography techniques which are usually applied to multimedia data hiding. In digital steganography, data are embedded in digital media, such as images or videos, and attempt not to be decrypted by others. So CDP-ABE can be treated as a steganographic encryption technology since the cover media is a ciphertext for someone while the data are contained in the ciphertext for the real receiver.

Our CDP-ABE construction follows the idea of *multi-distributional deniable encryption* which contains two sets of algorithms. One is claimed to be used while the other is actually used. The outputs of these two sets of algorithms need to be computationally indistinguishable or the deception will be easily discovered. Only the member who is authorized to use the deniable service can be aware of the existence of the deniable algorithm set.

The formal definition is presented as follows.

Definition 1 (Multi-Distributional CDP-ABE): A multi-distributional CDP-ABE scheme is composed of algorithms as follows:

1) **Setup**$(1^\lambda) \rightarrow \{\varepsilon, \mathcal{S}\}$: Given a security parameter λ, the algorithm generates a system-wise public information ε and a secret $\mathcal{S}$ which will be used for further key generation.
2) **KeyGen** $(\varepsilon, \mathcal{S}, S) \rightarrow SK_S$: Given the system-wise parameter ε, the secret $\mathcal{S}$ and a user attribute set S, this algorithm generates a secret key SK_S for the user.

3) **Enc**$(M, \varepsilon, \mathbb{A}) \rightarrow C$: Given a message M, the public information ε and the access policy $\mathbb{A}$, the algorithm encrypts M to a ciphertext C which can only be correctly decrypted by those who have the attributes that satisfy the given policy $\mathbb{A}$.
4) **Dec** $(C, SK_S) \rightarrow \{M, \perp\}$: Given a ciphertext C and a user's private key SK_S, the algorithm can correctly recover the original message M if S satisfies $\mathbb{A}$ defined in C. Otherwise, the algorithm simply replies $\perp$.
5) **OpenEnc**$(\varepsilon, C, M, \mathbb{A}) \rightarrow P$: This algorithm is for the sender to release encryption proof P to show that C is encrypted from M with the access policy $\mathbb{A}$.
6) **DenSetup**$(1^{\lambda}) \rightarrow \{\varepsilon, \mathbb{S}, \varepsilon', \mathbb{S}'\}$: The algorithm first runs **Setup** to obtain ε and $\mathbb{S}$. Then the algorithm generates ε' and $\mathbb{S}'$ for deniable use. Note that ε' is only known by the deniable-encryption service users and is kept secret from outsiders.
7) **DenKeyGen** $\left(\varepsilon, \mathbb{S}, \varepsilon', \mathbb{S}', S\right) \rightarrow \{SK_S, SK'_S\}$: This is used to generate keys for deniable users. SK_S is generated via **KeyGen**. The algorithm also generates another deniable key SK'_S. SK'_S can be used to decrypt both normal ciphertexts and deniable ciphertexts and can get the real message. As for SK_S, it can only be used to get the cover messages from the deniable ciphertexts. The existence of SK'_S is only known by the deniable service users.
8) **DenEnc** $\left(M, M', \varepsilon, \varepsilon', \mathbb{A}, \mathbb{A}'\right) \rightarrow C'$: Input two encryption tuples $M, \varepsilon, \mathbb{A}$, which is for the real message encryption, and $M', \varepsilon', \mathbb{A}'$, which is for the cover message encryption, the algorithm generates a ciphertext C'. Note that the output of **DecEnc** should be indistinguishable to the output of **Enc**. $\mathbb{A}'$ is called the cover policy in this paper.
9) **DenOpenEnc** $\left(\varepsilon, C', M', \mathbb{A}'\right) \rightarrow P'$: This algorithm is for the sender to release encryption proof P' to show that C is encrypted from M' with the access policy $\mathbb{A}'$. Note that P' should be computationally indistinguishable with P from **OpenEnc** or the outsider will learn the cheating fact.

It is obvious that the first four algorithms can be treated as a generic CP-ABE scheme. We call the first four algorithms the normal set of algorithms. **DenSetup**, **DenKeyGen** and **DenEnc** and **Dec**, constitute the primary characteristics of our proposal. We call them the deniable set of algorithms. Note that there is no **DenDec** algorithm since all entities should use the same decryption algorithm for ciphertext validation. There is no **OpenDec** which makes the receiver provide a proof because CDP-ABE misleads the coercer to other receivers instead of the real receiver group. So those innocent receivers can simply provide their keys to the coercer as the proofs. In our scheme, we allow all users to provide their keys from **KeyGen** and this will not crack the deniable feature.

The usage of CDP-ABE is described here. **DenSetup** is used to setup an operation environment. Though there are more outputs than **Setup**, the user can simply publish the normal part to outsiders and keep the additional part secret. **DenKeyGen** is used to generate a secret key pair, one is the normal secret key and the other is the deniable secret key. **DenEnc** is the deniable encryption function which can encrypt a real message and a cover message for different policies at the same time. Note that the output of **DenEnc** should be indistinguishable from the output of **Enc** with the same message and cover identity. This implies that the key of satisfying the cover policy can be used to decrypt C' and derive M', otherwise **Dec** would be a breakpoint. As for the deniable secret key, it is used to uncover the real message for the real receiver. In this way, the sender can claim that a published ciphertext is derived from **Enc** for the cover receiver while actually it is generated by **DenEnc** for the real receiver.

B. CDP-ABE PROPERTIES

In this subsection, we define the properties that a CDP-ABE scheme should satisfy. The *Correctness* and *Security* of the scheme are similar to those of traditional CP-ABE schemes. In addition to these two properties, the CDP-ABE must also satisfy *Receiver Indistinguishability*.

1) **Correctness**. In a CDP-ABE encryption scheme, correctness must be satisfied for both the cover receiver and the real receiver. That is, the following equations must be satisfied:
Dec(**Enc** $(M, \varepsilon, \mathbb{A})\, SK_S) \rightarrow M$, if S satisfies $\mathbb{A}$.
$$\textbf{Dec}(\textbf{DenEnc}\left(M, M', \varepsilon, \varepsilon', \mathbb{A}, \mathbb{A}'\right) SK'_S) \rightarrow M,$$
if S satisfies $\mathbb{A}$.
$$\textbf{Dec}(\textbf{DenEnc}\left(M, M', \varepsilon, \varepsilon', \mathbb{A}, \mathbb{A}'\right) SK_S) \rightarrow M,$$
if S satisfies $\mathbb{A}'$.
2) **Security**. The tuple {**Setup**, **KeyGen**, **Enc Dec**}must form a semantically-secure encryption scheme. The security model is defined as follows.
   a) **Setup**: The challenger runs **Setup** to create an encryption environment.
   b) **Phase 1**: The adversary generates key queries $q_1, \ldots, q_m$ for the challenger corresponding to attribute sets $S_1, \ldots, S_m$ and obtains their private keys.
   c) **Challenge**: The adversary chooses two plaintexts M_0, M_1 and an access policy $\mathbb{A}$ that it wants to be challenged by the challenger. The constraint is that $S_1, \ldots S_m$ cannot satisfy the access policy $\mathbb{A}$. The challenger randomly chooses one bit $b \in \{0, 1\}$ and encrypts the message via **Enc** $(M_b, \varepsilon, \mathbb{A}) \rightarrow C^*$. The challenger sends C^* back to the adversary.
   d) **Phase 2**: As in Phase 1, the adversary generates key queries $q_{m+1}, \ldots, q_{m+n}$ for a challenger corresponding to attribute sets $S_{m+1}, \ldots, S_{m+n}$ and obtains their private keys. Note that $S_{m+1}, \ldots, S_{m+n}$ cannot satisfy the access policy $\mathbb{A}$.
   e) **Guess**: The adversary returns the guess result $b' \in \{0, 1\}$. The adversary wins if $b' = b$.

We call a CDP-ABE scheme semantically secure if all polynomial time adversaries have at most a negligible advantage in the above game. Note that we do not care about the semantic security of **DenEnc**. This is because the outputs of **Enc** and **DenEnc** should be indistinguishable. If one is semantically secure while the other is not, the above game will distinguish if the claimed receiver is the cover receiver or the real receiver. This would violate the property below.

3) **Receiver Indistinguishability**. Indistinguishability achieves that outsiders cannot tell if the claimed receiver is the cover receiver or the real receiver. The formal definition is described as follows. Given a normally encrypted ciphertext C for an access group $\mathbb{A}$ and a deniably encrypted ciphertext C^* that is claimed to be for the access group $\mathbb{A}$, for every probabilistic polynomial time (PPT) algorithm A the advantage defined here

$$\mathrm{Adv}_A := \left|P\left[A\left(C^*\right)=1\right]-P\left[A\left(C\right)=1\right]\right|$$

is negligible.

C. CDP-ABE SCENARIO ASSUMPTION

In the above definition, there are two different user groups in this CDP-ABE service. One is the normal user group, where the member in this group is not aware of the deniability feature and simply treats the scheme as a common CP-ABE service. The other is the deniable user group, where the member can encrypt the message deniably and can get the real message under the cover ciphertext. Here we assume that the member in the deniable group will not be compromised. That is, the existence of ε, $\mathcal{S}$ and SK_S will not be released to the outsider. Since ABE is a kind of multicast encryption technique, if one valid receiver is compromised, undoubtedly all messages for this receiver will be leaked to the attacker and no protection mechanism works. So our assumption is reasonable.

IV. PRELIMINARIES

In this section, we introduce major techniques used in our schemes.

A. BILINEAR MAP GROUPS

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order p, with a map function $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. Let g be a generator of $\mathbb{G}$. $\mathbb{G}$ is a bilinear map group if $\mathbb{G}$ and e have the following properties:

1) **Bilinearity**: $\forall u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_n^*$, $e\left(u^a, v^b\right) = e(u, v)^{ab}$.
2) **Non-degeneracy**: $e(g, g) \neq 1$.
3) **Computability**: the group action in $\mathbb{G}$ and map function e can be computed efficiently.

B. COMPOSITE ORDER BILINEAR GROUPS

The composite order bilinear group was first introduced in [38]; we use it to construct our scheme. Here we provide a brief introduction. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of composite order $N = p_1p_2\ldots p_m$, where $p_1, p_2, \ldots, p_m$ are distinct primes, with bilinear map function $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. For each prime p_i, $\mathbb{G}$ has a subgroup $\mathbb{G}_{p_i}$ of order p_i. We let $g_1, g_2, \ldots, g_m$ be the generators of these subgroups respectively. Each element in $\mathbb{G}$ can be expressed in the form of $g_1^{a_1} g_2^{a_2} \ldots g_m^{a_m}$, where $a_1, a_2, \ldots, a_m \in \mathbb{Z}_N$. If a_i is congruent to zero modulo p_i, we say that this element has no $\mathbb{G}_{p_i}$ component. We say an element is in $\prod_{i\in S}\mathbb{G}_{p_i}$, where S is a subset from $1\ldots m$, if $\forall i{\in}S$, a_i is not congruent to zero modulo p_i.

Orthogonality between all subgroups under bilinear map e is one of the most important properties of the composite bilinear groups. Orthogonality means that if $u \in \mathbb{G}_{p_i}$, $v{\in}\mathbb{G}_{p_i}$ and $i \neq j$, then $e\left(u, v\right) = 1$, where 1 is the identity element in $\mathbb{G}_T$. In our scheme, we will use subgroups to create redundant spaces for different attribute sets.

We will also use the **subgroup decision assumption** of the composite order group. The assumption states that it is difficult to determine the existence of a given subgroup in a random composite order group element without orthogonality testing. The general form of this assumption is described as follows,

Definition 2 (General Subgroup Decision Assumption): Let $S_0, S_1, S_2, \ldots, S_k$ be non-empty subsets of $1\ldots,m$ such that for each $2 \leq j{\leq}k$, either $S_j \cap S_0 = \emptyset = S_j \cap S_1$ or $S_j \cap S_0 \neq \emptyset \neq S_j \cap S_1$. Given group generator $\mathcal{G}$, we define the following distribution:

$$\begin{aligned} PP &:= \{N = p_1p_2\ldots p_m, \mathbb{G}, \mathbb{G}_T, e\} \xleftarrow{R} \mathcal{G},\\ \mathcal{Z}_i &\xleftarrow{R} \mathbb{G}_{S_i}\forall i \in \{1, \ldots, k\},\\ D &:= PP, \mathcal{Z}_2, \ldots, \mathcal{Z}_k. \end{aligned}$$

We assume that for any PPT algorithm A with output in $\{0, 1\}$,

$$\mathrm{Adv}_{g,A} := \left|P\left[A\left(D, \mathcal{Z}_0\right)=1\right]-P\left[A\left(D, \mathcal{Z}_1\right)=1\right]\right|$$

is negligible.

This assumption implies that if all bilinear group members contain elements from at least one common subgroup, it is hard to tell the existence of elements from other subgroups.

C. CHAMELEON HASH

The chameleon hash scheme was first introduced by Krawczyk and Rabin [39]. Like other common secure hash functions, a chameleon hash scheme has two key properties, namely **collision resistance** and **semantic security**. Further, a chameleon hash scheme provides **collision forgery** with a pre-determined trapdoor. The input of a chameleon hash includes two parts: the input message m and a random string r. The random string r is used to provide a chance to adapt the message to the hash value. There are three phases in a chameleon hash scheme; each of them is summarized below.

1) **Setup**$(1\lambda) \rightarrow PK$: Given a security parameter λ, the scheme outputs a public parameter PK and a secret trapdoor SK.

2) **CH**$(m, r) \rightarrow H$: An efficient and probabilistic algorithm, with inputs of a message m, and a random number r, outputs a hash value H. PK is treated as an environment parameter, and we omit it in the inputs for simplicity.
3) **Forgery**$(SK, H, m') \rightarrow r'$: An efficient and probabilistic algorithm, with a given hash value H, a message m', and SK, outputs a random string r' that matches the hash value and the hash function.

The definitions of the three aforementioned requirements, **collision resistance**, **semantic security** and **collision forgery**, are listed below.

Definition 3 (Collision Resistance): Given a chameleon hash scheme $PK, SK, CH(\cdot, \cdot)$, where PK is the public information, SK is the trapdoor and $CH(\cdot, \cdot)$ is the hash function, let m, m' be two different messages and let r be a random string. We call the scheme **collision resistant** if for any PPT algorithm A, it is hard to output an r' such that $CH\,(m, r) = CH(m'r')$ without SK.

Definition 4 (Semantic Security): Given a chameleon hash scheme $PK, SK, CH(\cdot, \cdot)$, where PK is the public information, SK is the trapdoor and $CH(\cdot, \cdot)$ is the hash function, we call the scheme **semantically secure** if for all pairs of messages m, m' and random string r, the probability distribution of $CH\,(m, r)$ and $CH(m'r')$ are computationally indistinguishable.

Definition 5 (Collision Forgery): Given a chameleon hash scheme $PK, SK, CH(\cdot, \cdot)$, where PK is the public information, SK is the trapdoor and $CH(\cdot, \cdot)$ is the hash function, let m, m' be two different messages and r be a random string. We call the scheme a **collision forgery** scheme if there exists one PPT algorithm A that with an input of SK, outputs a string r' that satisfies $CH\,(m, r) = CH(m', r')$.

In this paper, we use CH to denote the chameleon hash public information and $CH(\cdot, \cdot)$ to denote the chameleon hash operation.

D. WATERS CP-ABE

We use the Waters ciphertext-policy attribute-based encryption (CP-ABE) scheme [40] to construct our steganographic ABE. In this subsection, we provide an introduction to the Waters CP-ABE. Waters used a Linear Secret Sharing Scheme (LSSS) to build an access control mechanism. The definition of LSSS is briefly described here.

Definition 6 (LSSS: Linear Secret Sharing Scheme [41]): A secret sharing scheme Π over a set of parties $\mathcal{P}$ is called linear (over $\mathcal{Z}_p$) if the following is true.

(a) The shares for each party form a vector over $\mathcal{Z}_p$.
(b) There exists an $l \times n$ matrix M called the share-generating matrix for Π. For all i= $1, \ldots, l$, the ith row of M is labelled by the party $\rho(i)$, where ρ is a mapping function from $\{1 \ldots, l\}$ to the party field $\mathcal{P}$. When considering a column vector $v = (s, r_2, \ldots, r_n)$, where $s \in \mathcal{Z}_p$ is the secret to be shared and $r_2, \ldots, r_n \in \mathcal{Z}_p$ are randomly chosen, Mv is a vector of l shares of secret s according to Π. The share $(Mv)_i$ belongs to party $\rho(i)$.

According to the above definition, an LSSS scheme has linear reconstruction property. That is, given LSSS Π, access structure $\mathbb{A} = (\mathcal{M}, \rho)$, and valid shares of a secret s, s can be recovered by those who have authorized sets. Beimel [41] demonstrated that the recovery procedure is a time polynomial in the size of M. In an ABE scheme, parties represent attributes; therefore, authorized sets imply groups with the required attributes. The Waters CP-ABE scheme is composed of the following algorithms.

1) **Setup**$(1^{\lambda}) \rightarrow \{\varepsilon, \mathcal{S}\}$: This algorithm chooses a bilinear group of prime order p with generator g, random elements $\alpha, a \in \mathcal{Z}_p$, and hash function $H : \{0, 1\}^* \rightarrow \mathbb{G}$. The system-wise public parameter PK is $\{g, e(g, g)^{\alpha} g^{a}\}$ and the system secret key MSK is g^{α}.
2) **Encrypt**$(PK\,(\mathcal{M}, \rho)\,, M) \rightarrow C$: Given message M and LSSS access structure $(\mathcal{M}, \rho)$, this algorithm first chooses a random vector $\vec{v} = (s, y_2, \ldots, y_n) \in \mathcal{Z}_p^n$. Let $\mathcal{M}$ be an $l \times n$ matrix and $\mathcal{M}_i$ denote the ith row of $\mathcal{M}$. This algorithm calculates $\lambda_i = \vec{v}\mathcal{M}_i, \forall i \in \{1 \ldots, l\}$. The output ciphertext is:
$$C = \Big\{Me\,(g, g)^{\alpha s}, g^{s}, g^{a\lambda_1} H\,(\rho\,(1))^{-s}\,, \\ \ldots, g^{a\lambda_l} H\,(\rho\,(1))^{-s}\Big\} = \Big\{C, C', C_1,\ \ldots, C_l\Big\},$$
with a description of $(\mathcal{M}\rho)$.
3) **KeyGen**$\,(MSK, S) \rightarrow SK$: Given set S of attributes, this algorithm randomly chooses $t \in \mathcal{Z}_p$ and outputs the private key such that
$K = g^{\alpha+at} L = g^{t}, \forall x \in SK_x = H(x)^{t}.K$ is defined originally as D in [40].
4) **Decrypt**$\,(C, SK) \rightarrow M$: Suppose that S satisfies the access structure and let $I \subset \{1 \ldots, l\}$ be defined as $I = \{i : \rho(i) \in S\}$. This algorithm finds a set of constants $\{w_i \in \mathcal{Z}_p\}_{i \in I}$ such that $\sum_{i \in I} w_i \lambda_i = s$. The decryption algorithm computes
$$e(C'K)/(\prod_{i \in I} (e\,(C_i, L)\, e(C', K_{\rho(i)}))^{w_i}) = e(g, g)^{\alpha s}$$
and derives M from the ciphertext.

Waters CP-ABE scheme is CPA-secure if the decisional q-BDHE assumption holds. The decisional q-BDHE assumption is defined as follows:

Definition 7 (Decisional Bilinear Diffie-Hellman Exponent Assumption): Let $a, s \in \mathcal{Z}_p$ and g be a generator of $\mathbb{G}$. Let g_i denote g^{a^i}. Given
$$D := \{g, g^{s}, g_1, \ldots, g_q, g_{q+2}, \ldots, g_{2q}\}$$
and an element $T \in \mathbb{G}_T$, we assume that for any PPT algorithm A that outputs in $\{0, 1\}$,
$$Adv_A := \left|P\left[A\left(D, e(g, g)^{a^{q+1}s}\right) = 1\right] - P[A\,(D, T) = 1]\right|$$
is negligible. The proof can be found in [40] and is skipped here.

V. CIPHERTEXT-DENIABLE-POLICY ATTRIBUTE-BASED ENCRYPTION

In this section, we build a CDP-ABE based on the Waters ABE scheme [40]. First, we address some design issues and propose our solutions for these problems. Then, we construct our CDP-ABE scheme in section V-*B*. In section V-*C*, we prove the correctness of our scheme. The security proof and the indistinguishability proof are in section V-*D* and V-*E* respectively. We evaluate the performance of our scheme in section V-*F*.

A. CONCEPT

Our CDP-ABE provides an important feature that the sender can deny the access policy contained in a ciphertext. Our CDP-ABE applies Waters scheme as the base scheme and therefore, we also use LSSS as the access structure to present the access policy. LSSS can be divided into two parts, one is the secret shares, which can be represented as a matrix $\mathbb{M}$, and the other is the mapping function ρ. ρ is used to present the relation between attributes and secret shares. Here we use the pre-determined deniable encryption technique to encrypt the mapping functions of both the cover policy and the real policy at the same time. So the encrypted mapping function can be opened to the real one or the cover one according to the receiver key. We use the composite order bilinear groups to develop the deniable encryption technique. A composite order bilinear group can be treated as a composition of multiple subgroups. We use these subgroups to create redundant spaces and put different mapping functions to different subgroups. We also use the collision forgery feature of the chameleon hash function to make both mapping functions convincing.

The second problem is about the secret shares. With different mapping functions, one secret share can be mapped to different attributes. How to interpret one share as different attributes is a big challenge. In Waters scheme, each attribute is bound with an attribute hash value. Only when the key's attribute is the same with the secret share, the attribute hash value part can be removed in the decryption process. Otherwise, the decryption result will be meaningless. Here we use the composite order bilinear group as the hash value. So hash values of two attributes can be stored in different subgroups. In our scheme, the only requirement of H' is a one way cryptographic hash function. A chameleon hash function without its trapdoor is a provable cryptographic function. That is, a user can use a chameleon hash function in the normal encryption function and there is no problem. So even an outsider finds out the ciphertext coming from a chameleon hash function, the user can claim that he uses the normal encryption instead of the deniable encryption. With the canceling properties, we ensure that the unwanted part can be canceled in the decryption process. Again, we use the collision forgery feature of the chameleon hash function to make two decryption results, which are from the normal key and the deniable key respectively, convincing.

B. CONSTRUCTION

We construct our CDP-ABE as follows:

1) **Setup**$(1^{\lambda}) \rightarrow \{\varepsilon, \mathcal{S}\}$: The algorithm generates bilinear group $\mathbb{G}$ of order $N = p_1 p_2 p_3$, where p_1, p_2, p_3 are distinct primes with a bilinear map function $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. $\mathbb{G}_T$ is also of order N. We use $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}, \mathbb{G}_{p_3}$ to denote three orthogonal subgroups in $\mathbb{G}$ of order p_1, p_2, p_3, respectively. The algorithm picks generators $g_1 \in \mathbb{G}_{p_1}$, $g_2 \in \mathbb{G}_{p_2}$, $g_3 \in \mathbb{G}_{p_3}$ random elements $a\alpha\beta \in \mathbb{Z}_p$ and a hash function $H : \{0, 1\}^* \rightarrow \mathbb{G}$. Then the algorithm prepares a hash function H_{12} that maps a random string to $\mathbb{G}_{p_1 p_2}$ and satisfies the following property:

$$e\left(H\left(x\right), g_1 g_2\right) = e\left(H_{12}\left(x\right), g_1 g_2\right), \quad \forall x$$

The system-wise parameter ε and the secret $\mathcal{S}$ are

$$\varepsilon = \left\{ \begin{array}{c} \mathbb{G}, \mathbb{G}_T, N, g_1 g_2, (g_1 g_2)^{a}, \\ (g_1 g_2)^{\beta}, e\left(g_1 g_1\right)^{\alpha}, e\left(g_2 g_2\right)^{\alpha}, \\ H_{12} \end{array} \right\},$$

$$S = \left\{ (g_1 g_2)^{\alpha}, H \right\}$$

2) **KeyGen** $(\varepsilon, \mathcal{S}, S) \rightarrow SK_S$: Given the system-wise parameter ε, the secret $\mathcal{S}$ and a set of S, the algorithm randomly chooses $t \in \mathbb{Z}_n$ and the output private key SK_S is

$$\begin{aligned} SK_S &= \left\{ (g_1 g_2)^{\alpha + at}, (g_1 g_2)^{t}, \{H_{12}\left(x\right)^{t}\}|_{\forall x \in S} \right\} \\ &= KL\{K_x\}\,|_{\forall x \in S} \end{aligned}$$

3) **Enc**$(M, \varepsilon, \mathbb{A} = \{\mathbb{M}, \rho) \rightarrow C$: Given a message M and the target LSSS access structure $\mathbb{A} = \mathbb{M}, \rho$, the algorithm encrypts the message for $\mathbb{A}$. First, the algorithm encrypts ρ as follows. The algorithm sets up a cryptographic one-way hash function H'. Note that the hash function H' can be any type of cryptographic one-way function and is determined during encryption and that a chameleon hash function can be applied here. The algorithm randomly picks $s_1 \in \mathbb{Z}_n$, two random strings $t_{0,0}, t_{0,1}$, and flips two coins b_0, b_1. The output will be

$$\Gamma = \{\varrho_0, \varrho_1, \varsigma, V_1, t_{0,0}, t_{0,1}\}$$

where,

$$\begin{aligned} \varrho_{b_0} &= \rho \cdot e\left(g_1, g_2, g_1 g_2\right)^{\beta_{s_1}}, \\ \varrho_{1-b_0} &\xleftarrow{R} \mathbb{G}_T, \\ \varsigma &= (g_1 g_2)^{s_1}, \\ V_1 &= H'\left(\rho, t_{0,b_1}\right) \\ &\neq H'\left(\varrho_{1-b_0} \cdot e\left(g_1 g_2, g_1 g_2\right)^{-\beta_{s_1}}, t_{0,1-b_1}\right). \end{aligned}$$

It is trivial that every user in this system can correctly derive ρ since $(g_1 g_2)^{\beta}$ is included in the public information ε.

Now the algorithm focuses on the attribute part. Let $\mathbb{M}$ be an $l \times n$ matrix and $\mathbb{M}_i$ denote the i-th row of $\mathbb{M}$. The algorithm first chooses two random vectors $\vec{v} = (s_2, y_2, \ldots, y_n) \in \mathbb{Z}_N^n$. This algorithm then calculates $\lambda_i = \vec{v}\mathbb{M}_i, \forall i \in \{1 \ldots, l\}$.

The algorithm flips another two coins b_2, b_3 and picks two random strings $t_{1,0}, t_{1,1}$. The output result is

$$\Delta = \{A_0, A_1, B, C_1, , \ldots, C_l, t_{1,0}, t_{1,1}, V_2\}$$

where,

$$\begin{aligned} A_{b_2} &= M \cdot e(g_1g_2, g_1g_2)^{\alpha s_2}, \\ A_{1-b_2} &= M \cdot e(g_1, g_1)^{\alpha s_2}, \\ B &= (g_1g_2)^{s_2}, \\ C_i &= (g_1g_2)^{a\lambda_i} H_{12}(\rho(i))^{-s_2}, \quad i = 1 \ldots l, \\ V_2 &= H'(M, t_{1,b_3}). \end{aligned}$$

The ciphertext C will be as follows:

$$C = \{\Gamma, \Delta, H'\mathbb{M}\}$$

4) **Dec** $(C, SK_S) \rightarrow \{M, \perp\}$: To decrypt a ciphertext C, the algorithm first computes possible ρ as follows.

$$\rho_i = \varrho_i \cdot e(\varsigma, (g_1g_2)^{\beta})^{-1}$$

The algorithm derives the correct ρ by checking

$$V_i \overset{?}{=} H'\left(\rho_i, t_{0,j}\right), \quad \forall i, j \in \{0, 1\},$$

If $H'(\rho_i, t_{0,1})$ is equal to V_i, we know ρ_i is the correct mapping function and b_0, b_1 are i, j respectively. If all candidates fail the equality verification, the algorithm replies $\perp$.

According to ρ and $\mathbb{M}$, the algorithm checks if the attribute set S of SK_S satisfies $\mathbb{A}$. If not, the algorithm returns $\perp$. Otherwise, let $I \subset \{1, 2, \ldots, l\}$ be defined as $I = \{i : \rho(i) \in S\}$. Then this algorithm finds a set of constants $\{w_i \in \mathcal{Z}_p\}$ such that $\sum_{i \in I} w_i \lambda_i = s_1$. This algorithm computes all possible messages as follows:

$$M_j = A_j \cdot \frac{e\left(\prod_{i \in I} C_i^{w_i}, L\right) e(B, \prod_{i \in I} K_{\rho(i)}^{w_i})}{e\left(B, K\right)}, \quad \forall j \in \{0, 1\}$$

The algorithm then verifies two candidates' messages with V by calculating

$$v_{j,k} = H'\left(M_j, t_{1,k}\right), \quad \forall j, k \in \{0, 1\}$$

If $v_{j,k}$ is equal to V_2, then M_j is the message and is returned. j, k are b_2, b_3, respectively, of coins selected by the encryptor. Otherwise, this algorithm returns $\perp$. Note that the most computationally intensive aspect of this algorithm,

which is $\frac{e\left(\prod_{i \in I} C_i^{w_i}, L\right) e(B, \prod_{i \in I} K_{\rho(i)}^{w_i})}{e(B,K)}$, only needs to be computed once. Therefore, the overall computation time compared to the base scheme will not be significantly larger.

5) **OpenEnc**$(\varepsilon, C, M, \mathbb{A}) \rightarrow P$: This algorithm returns the coins used in the encryption process b_0, b_1, b_2, b_3 as the proof P.

6) **DenSetup**$(1^{\lambda}) \rightarrow \{\varepsilon, \mathcal{S}, \varepsilon', \mathcal{S}'\}$: The algorithm first runs **Setup** to obtain $\varepsilon, \mathcal{S}$. The deniable system-wise parameter ε' and the secret $\mathcal{S}$' are

$$\varepsilon' = \left\{ \begin{array}{c} g_2g_3, (g_2g_3)^{a}, e(g_3, g_3)^{\alpha}, \\ (g_1g_2g_3)^{\beta}, e(g_1, g_1)^{\beta}, \\ H_{13}, H_2, H_3 \end{array} \right\}$$

$$\mathcal{S}' = \left\{(g_1g_3)^{\alpha}\right\}.$$

7) **DenKeyGen** $\left(\varepsilon, \mathcal{S}, \varepsilon', \mathcal{S}', S\right) \rightarrow \{SK_S, SK'_S\}$: Given an attribute set S, SK_S is directly generated by **KeyGen**. The deniable private key SK'_S is defined as follows:

$$\begin{aligned} SK'_S &= \left\{(g_1g_3)^{\alpha + at}, (g_1g_3)^{t}, \left\{H_{13}(x)^{t}\right\} |_{\forall x \in S}\right\} \\ &= \{K'L'\left\{K'_x\right\} |_{\forall x \in S} \end{aligned}$$

8) **DenEnc** $\left(M, \varepsilon, M', \varepsilon', \mathbb{A} = \{\mathbb{M}', \rho\}, \mathbb{A}' = \{\mathbb{M}', \rho'\}\right) \rightarrow C'$: Given the real message M with the access policy $\mathbb{A}$ and the cover message M' with the cover policy $\mathbb{A}$', the algorithm first deniably encrypts ρ, ρ' as follows. The algorithm sets up a chameleon hash function CH. The algorithm then randomly picks $s_1 \in \mathcal{Z}_N$, a random string $t_{0,0}$ and flips two coins b_0, b_1. The output will be

$$\Gamma' = \{\varrho'_0, \varrho'_1, \varsigma', V_1, t_{0,0}, t_{0,1}\}$$

where,

$$\begin{aligned} \varrho'_{b_0} &= \rho \cdot e\left(g_1g_3, g_1g_3\right)^{\beta_{s_1}}, \\ \varrho'_{1-b_0} &= \rho' \cdot e\left(g_1, g_1\right)^{\beta_{s_1}}, \\ \varsigma' &= (g_1g_3)^{s_1}, \\ V_1 &= CH\left(\rho, t_{0,b_1}\right) = CH(\rho' t_{0,1-b_1}). \end{aligned}$$

$t_{0,1}$ is a string generated from the chameleon hash function forgery.

Next, the algorithm processes the attribute part. For simplicity, we assume that the required attribute numbers to recover secrets from $\mathbb{A}$ and $\mathbb{A}$' are the same. That is, $\mathbb{M} = \mathbb{M}$'. We will remove this constraint in section VI. Let $\mathbb{M}$ be an $l \times n$ matrix and $\mathbb{M}_i$ denote the i-th row of $\mathbb{M}$. The algorithm first chooses two random vectors $\vec{v} = (s_2, y_2, \ldots, y_n) \in \mathcal{Z}_N^n$. This algorithm then calculates $\lambda_i = \vec{v}\mathbb{M}_i, \forall i \in \{1 \ldots, l\}$. The algorithm flips another two coins b_2, b_3 and picks a random string $t_{1,0}$. The output result is

$$\Delta' = \{A'_0, A'_1, B', C'_1, \ldots, C'_l, t_{1,0}, t_{1,1}, V_2\}$$

where,

$$\begin{aligned} A'_{b_2} &= M \cdot e(g_3, g_3)^{\alpha s_2}, \\ A'_{1-b_2} &= M' \cdot e(g_2, g_2)^{\alpha s_2}, \\ B' &= (g_2g_3)^{s_2}, \\ C'_i &= (g_2g_3)^{a\lambda_i}(H_2(\rho'(i)H_3(\rho(i))^{-s_2}, \quad i = 1 \ldots l, \\ V_2 &= CH\left(M, t_{1,b_3}\right) = CH\left(M', t_{1-b_3}\right). \end{aligned}$$

The ciphertext C' will be as follows:

$$C' = \{\Gamma', \Delta', CH, \mathbb{M}\}$$

9) **DenOpenEnc** $(\varepsilon, C, M, \mathbb{A}) \rightarrow P$: This algorithm returns the opposite coins used in the encryption process $1 - b_0, 1 - b_1, 1 - b_2, 1 - b_3$ as the proof P'.

In our design, the normal algorithm set runs on $\mathbb{G}_{p_1p_2}$. As for the deniable algorithm set, the key is on $\mathbb{G}_{p_1p_3}$ while the ciphertext is on $\mathbb{G}_{p_2p_2}$. So in the decryption process, users will get different messages with different keys. The detail is shown in section III. The sender proof is the four random bits used in the encryption process.

C. CORRECTNESS

Here, we show the correctness of our CDP-ABE scheme. There are four cases that need to be checked. The first case is that a normally encrypted ciphertext can be correctly decrypted by the normal private key for the normal receiver group. The second case is that a normally encrypted ciphertext can be correctly decrypted by the deniable private key for the deniable receiver group. The third case is that a deniably encrypted ciphertext can be correctly decrypted by the deniable private key for the deniable receiver group and the real message is derived. The fourth case is that a deniably encrypted ciphertext can be correctly decrypted by the deniable private key for the normal receiver group and the fake message is derived.

1) **Normally encrypted ciphertext and the normal key**: First, the user can use ε to calculate

$$\begin{aligned} e\left(\varsigma, (g_1g_2)^{\beta}\right) &= e\left((g_1g_2)^{s_1}, (g_1g_1)^{\beta}\right) \\ &= e(g_1g_2, g_1g_2)^{\beta s_1}. \end{aligned}$$

By the V_1 verification with the hash function H', the normal public information ε can be used to correctly derive ρ. So the access policy $\mathbb{A}$ is opened. Then we focus on the message part. Because

$$\begin{aligned} &\frac{e\left(\prod_{i\in I} C_i^{w_i}, L\right) e(B, \prod_{i\in I} K_{\rho(i)}^{w_i})}{e\left(B, K\right)} \\ &= e(\prod\nolimits_{i\in I} (g_1g_2)^{a\lambda_i w_i} H_{12}(\rho(i))^{-s_2 w_i}, (g_1g_2)^t) \\ &\quad \cdot e((g_1g_2)^{s_2}, \prod\nolimits_{i\in I} H_{12}(\rho(i))^{tw_i}) \\ &\quad \cdot e((g_1g_2)^{s_2}, (g_1g_2)^{\alpha+at})^{-1} \\ &= e(\prod\nolimits_{i\in I} (g_1g_2)^{a\lambda_i w_i}, (g_1g_2)^t) \\ &\quad \cdot e((g_1g_2)^{s_2}, (g_1g_2)^{\alpha+at})^{-1} \\ &= e((g_1g_2)^{as_2}, (g_1g_2)^t) \\ &\quad \cdot e((g_1g_2)^{s_2}, (g_1g_2)^{\alpha+at})^{-1} \\ &= e(g_1g_2g_3g_4)^{-\alpha s_2} \end{aligned}$$

with the hash function H' and the verification tag V_2, the receiver can derive the message M.

2) **Deniably encrypted ciphertext and the normal key**: Since the normal user will not know the ciphertext is generated from deniable encryption, it uses ε to calculate

$$e\left(\varsigma', (g_1g_2)^{\beta}\right) = e\left((g_1g_3)^{s_1}, (g_1g_2)^{\beta}\right) = e(g_1, g_1)^{\beta s_1}$$

By the V_1 verification with the chameleon hash function CH, the normal scheme user can correctly derive ρ'. So the cover access policy $\mathbb{A}$' is opened. Then we focus on the message part. Because

$$\begin{aligned} &\frac{e\left(\prod_{i\in I} C_i^{'w_i}, L\right) e(B', \prod_{i\in I} K_{\rho(i)}^{w_i})}{e\left(B', K\right)} \\ &= e(\prod\nolimits_{i\in I}(g_2g_3)^{a\lambda_i w_i}(H_2(\rho'(i))H_3(\rho(i)))^{-s_2 w_i},(g_1g_2)^t) \\ &\quad \cdot e((g_2g_3)^{s_2}, \prod\nolimits_{i\in I} H_{12}(\rho'(i))^{tw_i}) \\ &\quad \cdot e((g_2g_3)^{s_2}, (g_1g_2)^{\alpha+at})^{-1} \\ &= e(\prod\nolimits_{i\in I} g_2^{a\lambda_i w_i}(H_2(\rho'(i)))^{-s_2 w_i} g_2^t) \\ &\quad \cdot e(g_2^{s_2}, \prod\nolimits_{i\in I} H_2(\rho'(i))^{tw_i}) \\ &\quad \cdot e(g_2^{s_2}, g_2^{\alpha+at})^{-1} \\ &= e(\prod\nolimits_{i\in I} g_2^{a\lambda_i w_i}, g_2^t)\cdot e(g_2^{s_2}, g_2^{\alpha+at})^{-1} \\ &= e(g_2^{as_2}, g_2^t)\cdot e(g_2^{s_2}, g_2^{\alpha+at})^{-1} \\ &= e(g_2, g_2)^{-\alpha s_2} \end{aligned}$$

with the chameleon hash function CH and the verification tag V_2, the normal receiver can derive the cover message M'.

3) **Deniably encrypted ciphertext and the deniable key**: Though the deniable user does not know if the ciphertext is normally encrypted or not, it just uses ε' to calculate

$$\begin{aligned} e\left(\varsigma', (g_1g_2g_3)^{\beta}\right) &= e\left((g_1g_3)^{s_1}, (g_1g_2g_3)^{\beta}\right) \\ &= e(g_1g_3, g_1g_3)^{\beta s_1} \end{aligned}$$

By the V_1 verification with the chameleon hash function CH, the deniable service user can correctly derive ρ. So the real access policy $\mathbb{A}$ is opened. Then we focus on the message part. Because

$$\begin{aligned} &\frac{e\left(\prod_{i\in I} C_i^{'w_i}, L'\right) e(B', \prod_{i\in I} K_{\rho(i)}^{'w_i})}{e\left(B', K'\right)} \\ &= e(\prod\nolimits_{i\in I}(g_2g_3)^{a\lambda_i w_i}(H_2(\rho'(i))H_3(\rho(i)))^{-s_2 w_i},(g_1g_3)^t) \\ &\quad \cdot e((g_2g_3)^{s_2}, \prod\nolimits_{i\in I} H_{13}(\rho(i))^{tw_i}) \\ &\quad \cdot e((g_2g_3)^{s_2}, (g_1g_3)^{\alpha+at})^{-1} \\ &= e(\prod\nolimits_{i\in I} g_3^{a\lambda_i w_i}(H_3(\rho'(i)))^{-s_2 w_i}, g_3^t) \\ &\quad \cdot e(g_3^{s_2} \prod\nolimits_{i\in I} H_3(\rho'(i))^{tw_i}) \\ &\quad \cdot e(g_3^{s_2}, g_3^{\alpha+at})^{-1} \\ &= e(\prod\nolimits_{i\in I} g_3^{a\lambda_i w_i}, g_3^t)\cdot e(g_3^{s_2}, g_3^{\alpha+at})^{-1} \\ &= e(g_3^{as_2}, g_3^t)\cdot e(g_3^{s_2}, g_3^{\alpha+at})^{-1} \\ &= e(g_3, g_3)^{-\alpha s_2} \end{aligned}$$

with the chameleon hash function CH and the verification tag V_2, the normal receiver can derive the real message M.

4) **Normally encrypted ciphertext and the deniable key**: Though the deniable user does not know if the ciphertext is normally encrypted or not, it just uses ε' to calculate

$$\begin{aligned} e\left(\varsigma, (g_1g_2g_3)^{\beta}\right) &= e\left((g_1g_2)^{s_1}, (g_1g_2g_3)^{\beta}\right) \\ &= e(g_1g_2, g_1g_2)^{\beta s_1} \end{aligned}$$

By the V_1 verification with the chameleon hash function H', the deniable scheme user can correctly derive ρ. So the real access policy $\mathbb{A}$ is opened. Then we focus on the message part. Because

$$\begin{aligned}&\frac{e\left(\prod_{i\in I} C_i^{w_i}, L'\right) e(B, \prod_{i\in I} K'^{w_i}_{\rho(i)})}{e\left(B, K'\right)}\\ &= e(\prod_{i\in I} (g_1g_2)^{a\lambda_i w_i} H_{12}\rho(i))^{-s_2w_i}, (g_1g_3)^t)\\ &\quad \cdot e((g_1g_2)^{s_2}, \prod_{i\in I} H_{13}(\rho(i))^{tw_i})\\ &\quad \cdot e((g_1g_2)^{s_2}, (g_1g_3)^{\alpha+at})^{-1}\\ &= e(\prod_{i\in I} g_1^{a\lambda_i w_i}, g_1^t) \cdot e(g_1^{s_2}, g_1^{\alpha+at})^{-1}\\ &= e(g_1^{as_2}g_1^t)\cdot e(g_1^{s_2}, g_1^{\alpha+at})^{-1} = e(g_1, g_1)^{-\alpha s_2}\end{aligned}$$

With the hash function H' and the verification tag V_2, the receiver can derive the message M.

From the above verification, we can find that the deniable user does not need to know if a message is normally encrypted or not. The deniable user only needs to use ε' and SK'_S to get the correct message. When being forced to release the key, the deniable user can claim that its attributes do not satisfy the ciphertext's access policy and its key is SK_S.

D. SECURITY PROOF

Our CDP-ABE is composed of two sets of algorithms, one is the normal set and the other is the deniable set. We only prove the security of the normal algorithm set here. This is because the normally encrypted ciphertext and the deniably encrypted ciphertext are indistinguishable, which is proved in section V-*E*. If one is broken and the other is secure, it is easy to use this property to tell the ciphertext and it conflicts with their indistinguishability.

A ciphertext is a tuple with four elements, $\{\Gamma, \Delta, H'\mathbb{M}\}$. Since H', $\mathbb{M}$ are public, Γ is also public and every system user who has ε can derive ρ, we only focus on the security of $\Delta, \mathbb{A}$. To prove the security, we reduce Waters CP-ABE to the normal set of algorithms in our CDP-ABE scheme. Since all subgroups are orthogonal, we can change the query, the response and the challenge in Waters scheme to our CDP-ABE scheme. The formal proof is described as follows.

Theorem 1: Our proposed CDP-ABE scheme is CPA secure if the Waters CP-ABE is CPA secure.

Proof: Let $\mathcal{A}$ be an adversary that breaks the above CDP-ABE scheme. We can construct algorithm $\mathcal{B}$ that can break Waters CP-ABE as follows. Let $\mathcal{X}$ denote the challenger of Waters scheme. $\mathcal{B}$ provides a group $\mathbb{G}_{p_1}$ with a prime p_1 and a hash function H_1 to $\mathcal{X}$. Then $\mathcal{B}$ is given public parameters through the Waters CP-ABE scheme's **Setup** algorithm from challenger $\mathcal{X}$

$$\varepsilon_{p_1} := \left\{g_1, g_1^{a_1}, e(g_1, g_1)^{\alpha_1}\right\}$$

For convenience, we use the suffix to represent different subgroups in our proof. Algorithm $\mathcal{B}$ proceeds as follows.

1) **Setup**: $\mathcal{B}$ first picks two different prime numbers p_2 and p_3. $\mathcal{B}$ generates group $\mathbb{G}$ with order $N = p_1p_2p_3$. Note that the subgroup with p_1 order in $\mathbb{G}$ should be the same as G_{p_1}. $\mathcal{B}$ sets up $\mathbb{E}_{p_2}$ with the Waters CP-ABE **Setup** algorithm from $\mathbb{G}_{p_2}$ and outputs $\{g_2, g_2^{a_2}, e\,(g_2, g_2)^{\alpha_2}\}$, where a_2, α_2 are in $\mathcal{Z}_{p_2}$. $\mathcal{B}$ randomly picks $\beta \in \mathcal{Z}_N$. Next, $\mathcal{B}$ shows

$$\varepsilon = \{\mathbb{G}, \mathbb{G}_T, N, g_1g_2, g_1^{a_1}, g_2^{a_2}(g_1g_2)^{\beta}, e(g_1, g_1)^{\alpha_1},\\ e(g_2, g_2)^{\alpha_2}H_{12}\}$$

to $\mathcal{A}$. Note that $\mathcal{B}$ is the one who knows p_1, p_2, p_3, so it is easy for $\mathcal{B}$ to generate required hash functions. Though a_1 is secret and different from a_2, which comes from $\mathcal{Z}_{p_1}$ and $\mathcal{Z}_{p_2}$ respectively, $g_1^{a_1}g_2^{a_2}$ can be treated as $(g_1g_2)^a$, where $a \in \mathcal{Z}_N$ from the Chinese remainder theorem. For the same reason, $e(g_1, g_1)^{\alpha_1}e(g_2, g_2)^{\alpha_2}$ can be treated as $e(g_1g_2, g_1g_2)^{\alpha}$, where $\alpha \in \mathcal{Z}_N$.

2) **Phase 1**: When $\mathcal{B}$ receives a key generation query for attribute set S from $\mathcal{A}$, $\mathcal{B}$ simply relays the query to $\mathcal{X}$ and obtains $SK_{p_1} = \{K_{p_1}L_{p_1}, \{K_{x,p_1}\}_{\forall x \in S}\}$. $\{K_{x,p_1}\}$ implies $H_1(x)^t$. $\mathcal{B}$ generates $\{K_{p_2}, L_{p_2}, \{K_{x,p_2}\}_{\forall x \in S}\}$ with the same algorithm. Again, $\mathcal{B}$ does not need to know the secret $\mathcal{X}$ uses. Next, $\mathcal{B}$ replies $\mathcal{A}$ the secret key SK as follows:

$$SK = \left\{K_{p_1}K_{p_2}, L_{p_1}L_{p_2}, \left\{K_{x,p_1}K_{x,p_2}\right\}_{\forall x \in S}\right\}$$

3) **Challenge**: $\mathcal{A}$ outputs two messages $\mathcal{M}_0, \mathcal{M}_1$ with access structure $(\mathbb{M}, \rho)$ to $\mathcal{B}$. $\mathcal{B}$ directly relays $\mathcal{M}_0, \mathcal{M}_1$ and $(\mathbb{M}, \rho)$ to $\mathcal{X}$ as the challenge and obtains

$$\{\mathcal{M}^* \cdot e\,(g_1, g_1)^{\alpha_1 s_1}, g_1^{s_1}, g_1^{a_1\lambda_i}H_1(\rho(i))^{-s_1}, i{=}\,1\ldots l\}$$

from $\mathcal{X}$. $\mathcal{M}^* \in \{M_0, M_1\}$ is chosen by $\mathcal{X}$. $\mathcal{B}$ setups a chameleon hash function CH and randomly picks b_0, b_1 from $\{0,\ 1\}$, s_2 from $\mathcal{Z}_{p_2}$. $\mathcal{B}$ also calculates $\{\lambda'_1, \ldots, \lambda'_l\}$. Finally, $\mathcal{B}$ outputs C to $\mathcal{A}$ as follows:

$$C = \{A_0, A_1, B, C_1, \ldots, C_l, CH, t_0, t_1, V\}$$

where

$$\begin{aligned}A_{b_0} &= M^* \cdot e(g_1, g_1)^{\alpha_1 s_1}e(g_2, g_2)^{\alpha_2 s_2},\\ A_{1-b_0} &= M^* \cdot e(g_1, g_1)^{\alpha_1 s_1},\\ B &= g_1^{s_1} \cdot g_2^{s_2},\\ C_i &= g_1^{a_1\lambda_i} \cdot g_2^{a_2\lambda'_i}H_1\left(\rho\left(i\right)\right)^{-s_1}H_2\left(\rho\left(i\right)\right)^{-s_2}\\ &\qquad\qquad \forall i = 1\ldots l,\\ V &= CH\left(\mathcal{M}_0, t_{b_1}\right) = CH\left(\mathcal{M}_1, t_{1-b_1}\right).\end{aligned}$$

Because of the Chinese remainder theorem, $\mathcal{A}$ will treat C as a ciphertext that comes from secret $s \in \mathcal{Z}_N$. Here, a chameleon hash function is used instead of a normal hash function; however, to $\mathcal{A}$, who has no trapdoor for the chameleon hash function, the chameleon hash function is just a normal one-way hash function.

4) **Phase 2**: $\mathcal{A}$ submits key generation queries to $\mathcal{B}$ and $\mathcal{B}$ responds as shown in **Phase 1**.

5) **Guess**: Finally, adversary $\mathcal{A}$ outputs guess b' to $\mathcal{B}$ and $\mathcal{B}$ uses b' to reply $\mathcal{X}$.

If $\mathcal{A}$ achieves a non-negligible advantage against the deniable scheme from our construction, $\mathfrak{B}$ can use the output of $\mathcal{A}$ to also achieve a non-negligible advantage against the Waters ABE scheme in the CPA model.

Since Waters CP-ABE scheme is CPA-secure if the decisional q-BDHE assumption holds, we can have the following theorem.

Theorem 2: Our proposed CDP-ABE scheme is CPA secure if the decisional q-BDHE assumption holds.

E. INDISTINGUISHABILITY PROOF

In this subsection, we prove that the output from the normal set of algorithms should be indistinguishable from the output from the deniable set of algorithms. We only focus on the ciphertext indistinguishability. As for the user key, since the opened key from the deniable user is generated from **KeyGen**, which is the same with the normal user, there is no indistinguishability issue.

A ciphertext is composed by $\Gamma, \Delta, H, \mathbb{M}$. Since a chameleon hash function is definitely a valid cryptographic one-way hash function, we cannot use this as a point to differentiate the kinds of ciphertexts. There is no difference about the access matrix $\mathbb{M}$. As for Γ and Δ, $t_{0,0}, t_{0,1}, t_{1,0}, t_{1,1}$ are random strings (though two of them are generated from the chameleon hash function, without the trapdoor they look like two random strings.) and V_1, V_2 are hash values. A_0, A_1 are proved to be secure, which implies computationally indistinguishable to a random element, and ϱ_0, ϱ_1 are the same case. So the difference between a normal ciphertext and a deniable ciphertext can be reduced to the problem of finding the difference between the following two tuples:

$$\mathbb{C} = \{(g_1g_2)^{s_1}\ (g_1g_2)^{s_2}, \{(g_1g_2)_i^{a\lambda}H_{12}(\rho(i))^{-s_2}\}\}$$

and

$$\mathbb{C}' = \left\{(g_1g_3)^{s_1}, (g_2g_3)^{s_2}, \left\{(g_2g_3)^{a\lambda_i} \left(H_2\left(\rho^{'(i)}\right) H_3\left(\rho\left(i\right)\right)\right)^{-s_2}\right\}\right\}.$$

So if $\mathbb{C}$ and $\mathbb{C}$' are indistinguishable, the normal ciphertext and the deniable ciphertext are indistinguishable, too. We setup some intermediate tuples for the proof use as follows.

$$\mathbb{C}_1 = \{(g_1g_3)^{s_1}, (g_1g_2)^{s_2}, \{(g_1g_2)^{a\lambda_i}H_{12}(\rho(i))^{-s_2}\}\}$$
$$\mathbb{C}_2 = \{(g_1g_3)^{s_1}, (g_1g_2)^{s_2}, \{(g_2g_3)^{a\lambda_i}H_{12}(\rho(i))^{-s_2}\}\}$$

Lemma 1: Under the general subgroup decision assumption, $\mathbb{C}$ and $\mathbb{C}_1$ are computationally indistinguishable.

Proof: We suppose there exists PPT attacker $\mathcal{A}$ who achieves a non-negligible advantage in distinguishing $\mathbb{C}$ from $\mathbb{C}_1$. We can create PPT algorithm $\mathfrak{B}$ that has a non-negligible advantage against the general subgroup decision assumption.

$\mathfrak{B}$ receives $N = p_1p_2p_3, g_1g_2T$, where g_1, g_2 belong to $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}$ respectively. $\mathfrak{B}$ wants to know if T belongs to G_{p_1,p_2} or G_{p_1,p_3}. $\mathfrak{B}$ randomly picks $\alpha\beta a \in \mathbb{Z}_N$ and setups the public information ε. $H_{12}(x)$ is defined as $(g_1g_2)^{h(x)}$ where h is a hash function mapping from a random string to $\mathbb{Z}_N$. $\mathfrak{B}$ publishes ε to $\mathcal{A}$. When receiving the key queries from $\mathcal{A}$, $\mathfrak{B}$ simply runs **KeyGen** and replies the generated keys. Then $\mathcal{A}$ sends an encryption challenge to $\mathfrak{B}$ with an access structure $\mathbb{A} = \{\mathbb{M}, \rho\}$. $\mathfrak{B}$ calculates $\{\lambda_i\}$ and returns the following tuple:

$$\mathbb{C}^* = \{T, (g_1g_2)^{s_2}\{(g_1g_2)^{a\lambda_i}H_{12}(\rho(i))^{-s_2}\}\}$$

If $T \in \mathbb{G}_{p_1p_2}$, then $\mathbb{C}^* \in \mathbb{C}$; otherwise, $\mathbb{C}^* \in \mathbb{C}_1$. If $\mathcal{A}$ has a non-negligible advantage over the tuple decision problem, $\mathfrak{B}$ can also have a non-negligible advantage over the subgroup decision problem.

Next, we want to prove $\mathbb{C}_1$ and $\mathbb{C}_2$ are indistinguishable.

Lemma 2: Under the general subgroup decision assumption, $\mathbb{C}_1$ and $\mathbb{C}_2$ are computationally indistinguishable.

Proof: We suppose there exists PPT attacker $\mathcal{A}$ who achieves a non-negligible advantage in distinguishing $\mathbb{C}_1$ from $\mathbb{C}_2$. We can create PPT algorithm $\mathfrak{B}$ that has a non-negligible advantage against the general subgroup decision assumption.

$\mathfrak{B}$ receives $N = p_1p_2p_3, g_1g_2, g_1g_3, T$, where g_1, g_2, g_3 belong to $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}, \mathbb{G}_{p_3}$ respectively. $\mathfrak{B}$ wants to know if T belongs to $\mathbb{G}_{p_1,p_2}$ or $\mathbb{G}_{p_2,p_3}$. $\mathfrak{B}$ randomly picks $\alpha, \beta, a \in \mathbb{Z}_N$ and setups the public information ε. $H_{12}(x)$ is defined as $(g_1g_2)^{h(x)}$ where h is a hash function mapping from a random string to $\mathbb{Z}_N$. $\mathfrak{B}$ publishes ε to $\mathcal{A}$. When receiving the key queries from $\mathcal{A}$, $\mathfrak{B}$ simply runs **KeyGen** and replies the generated keys. Then $\mathcal{A}$ sends an encryption challenge to $\mathfrak{B}$ with an access structure $\mathbb{A} = \{\mathbb{M}, \rho\}$. $\mathfrak{B}$ returns the following tuple:

$$\mathbb{C}^* = \{(g_1g_3)^{s_1}, (g_1g_2)^{s_2}\{T^aH_{12}(\rho(i))^{-s_2}\}\}$$

Note that $\{\lambda_i\}$ are not calculated since we assume that all elements in $\{\lambda_i\}$ are equal. If $T \in \mathbb{G}_{p_1p_2}$, then $\mathbb{C}^* \in \mathbb{C}_1$; otherwise, $\mathbb{C}^* \in \mathbb{C}_2$. If $\mathcal{A}$ has a non-negligible advantage over the tuple decision problem, $\mathfrak{B}$ can also have a non-negligible advantage over the subgroup decision problem.

Finally, we want to prove $\mathbb{C}_2$ and $\mathbb{C}$' are indistinguishable.

Lemma 3: Under the general subgroup decision assumption, $\mathbb{C}_2$ and $\mathbb{C}$' are computationally indistinguishable.

Proof: We suppose that there exists PPT attacker $\mathcal{A}$ who achieves a non-negligible advantage in distinguishing $\mathbb{C}_2$ from $\mathbb{C}$'. We can create PPT algorithm $\mathfrak{B}$ that has a non-negligible advantage against the general subgroup decision assumption.

$\mathfrak{B}$ receives $N = p_1p_2p_3, g_1g_2, g_1g_3, g_2g_3, T$, where g_1, g_2, g_3 belong to $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}, \mathbb{G}_{p_3}$ respectively. $\mathfrak{B}$ wants to know if T belongs to G_{p_1,p_2} or G_{p_2,p_3}. $\mathfrak{B}$ randomly picks $\alpha, \beta, a \in \mathbb{Z}_N$ and setups the public information ε. $H_{12}(x)$ is defined as $(g_1g_2)^{h(x)}$ where h is a hash function mapping from a random string to $\mathbb{Z}_N$. $\mathfrak{B}$ publishes ε to $\mathcal{A}$. When receiving the key queries from $\mathcal{A}$, $\mathfrak{B}$ simply runs **KeyGen** and replies the generated keys. Then $\mathcal{A}$ sends an encryption challenge to $\mathfrak{B}$ with an access structure $\mathbb{A} = \{\mathbb{M}, \rho\}$ and a cover access

TABLE 1. The group order of each scheme.

scheme	Group Order
Lai [4]	$p_1p_2p_3p_4$
Wang [5]	$p_1p_2p_3$
Zhang [7]	$p_1p_2p_3p_4$
Chi [14]	$p_1p_2p_3$
Ours (composite order)	$p_1p_2p_3$
Ours (simulation)	p

structure $\mathbb{A}' = \{\mathbb{M}', \rho'\}$. Again, we assume that $\mathbb{M} = \mathbb{M}'$. $\mathcal{B}$ calculates $\{\lambda_i\}$ and returns the following tuple:

$$\mathbb{C}^* = \{(g_1g_3)^{s_1}, T, \{(g_2g_3)^{a\lambda_i}T^{-h(\rho(i))}\}\}$$

If $T \in \mathbb{G}_{p_1p_2}$, then $\mathbb{C}^* \in \mathbb{C}_2$; otherwise, This construction implies that $h(\rho(i)) \equiv h\left(\rho'(i)\right) modp_3$ and therefore, $\mathbb{C}^* \in C'$. If $\mathcal{A}$ has a non-negligible advantage over the tuple decision problem, $\mathcal{B}$ can also have a non-negligible advantage over the subgroup decision problem.

According to the above lemmas, we can show that $\mathbb{C}$ and $\mathbb{C}$' are indistinguishable. So we can derive the following theorem.

Theorem 3: The normal ciphertext C and deniable ciphertext C' in our proposed CDP-ABE scheme are indistinguishable.

F. PERFORMANCE EVALUATION

Here we compare our construction with our CDP-ABE scheme with Waters' CP-ABE. The experiment environment is an INTEL i7-7700 computer. The chameleon hash function implementation follows [42]. Since we use the composite order group to construct scheme, the required computational time grows substantially. To solve this problem, we use the simulation technique proposed by Lewko [43]. Lewko formed a basis with some prime order group elements and made each base orthogonal to each other. So each basis can be treated as a composite order element. We implement our scheme with both the composite order group and the prime order group.

To our best of knowledge, there are no schemes which can provides target and message deniability at the same time. Therefore, we compare our scheme with some similar CP-ABE schemes which can hide the access policy from others. These schemes are called HCP-ABE. We compare our schemes with Lai *et al.* [4], Wang and He [5] and Zhang *et al.* [7]. We also compare our scheme with Chi's deniable CP-ABE scheme [14]. All these comparisons are based on the bilinear composite order group.

We summarize their order in table 1. Each prime size is 512 bits in our implementation. The comparison results are shown in Fig. 2 and Fig. 3.

From the comparison, we can find that the composite order scheme is much slower than the prime order scheme, as described above.

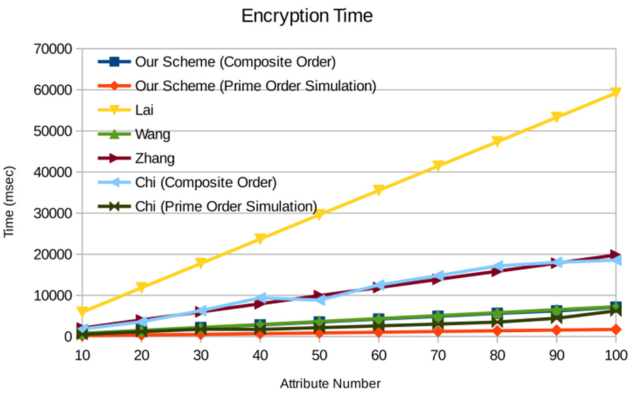

FIGURE 2. The encryption cost comparison with other schemes.

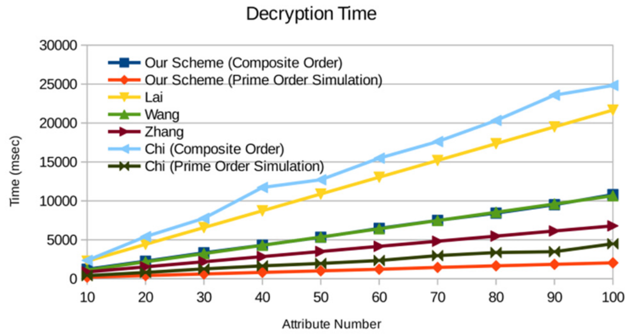

FIGURE 3. The decryption cost comparison with other schemes.

Our prime order simulation approach has the best performance in both encryption and decryption among these schemes. Moreover, our scheme provides the deniable access policy feature which is not included in these schemes. Although our CDP-ABE scheme is undoubtedly slower than Waters scheme which is our base scheme, we believe that the trade-off of our scheme is affordable.

VI. DISCUSSION: COVER ACCESS POLICY GENERATION ISSUE

In our construction, we assume that the secret recovery process is the same for the real access policy and the cover access policy. That is, given $\mathbb{A} = \{\mathbb{M}, \rho\}$ and $\mathbb{A}' = \{\mathbb{M}', \rho'\}$, we assume that $\mathbb{M} = \mathbb{M}'$. Though this requirement can be satisfied by the appropriate cover policy generation and is not impractical, we want to support arbitrary access policy in our scheme.

To solve this issue, we apply NOP (no operation) instruction concept in assembly. NOP is a CPU instruction with no function. It is often used in time alignment, memory alignment, hazard prevention, etc.. Refer to NOP, we setup many **null attributes** in our CDP-ABE system. All system users have keys for these null attributes. So when two policies have different secret composition, we can enlarge the smaller policy to the larger policy size and fill the additional required attributes with null attributes. Since every member has these null attribute keys, the target receiver group does not change.

For instance, suppose a real policy specifies that the receiver needs three attributes (**freshman**, **CS**, **scholarship**) while a plausible cover policy specifies that the receiver needs two attributes (**sophomore**, **EE**), the sender can enlarge the cover policy to the condition (**sophomore**, **EE**, **NULL**). This modification does not affect the number of the cover receivers. Undoubtedly, there can be many null attributes and it is not difficult to align the cover policy with the real policy. So in our scheme, we can say $\mathbb{M} = \mathbb{M}'$ without losing generalization.

VII. CONCLUSION AND FUTURE WORK

In this paper, we propose a CDP-ABE scheme. Compared to other CP-ABE schemes, CDP-ABE scheme makes a ciphertext include more than one access policy, one is the real policy and the other is the cover policy. We prove that the ciphertext with only one access group is computationally indistinguishable than the ciphertext with two access groups. So the existence of the real receiver group can be hidden under the receiver group. When being coerced, the sender can simply claim that the ciphertext is for the cover receiver group with proofs.

Our next step will focus on the issue of key management in CDP-ABE. In our schemes, users are divided into two groups; one group treats the scheme as a normal CP-ABE scheme while the other group enjoys the benefit of hiding receivers. The problem is the dynamic group member issue. If a user belonging to the deniable service group leaves the group, this member can know the real access policy of the ciphertexts. To address the issue, we consider enhancing our method by supporting a key management mechanism in the deniable service group.

REFERENCES

[1] R. Canetti, U. Feige, O. Goldreich, and M. Naor, "Adaptively secure multi-party computation," in *Proc. 28th Annu. ACM Symp. Theory Comput. (STOC)*, New York, NY, USA, 1996, pp. 639–648.

[2] R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky, "Deniable encryption," in *Proc. CRYPTO*, Santa Barbara, CA, USA, 1997, pp. 90–104.

[3] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *Applied Cryptography and Network Security*, S. M. Bellovin, R. Gennaro, A. Keromytis M. Yung, Eds. Berlin, Germany: Springer, 2008, pp. 111–129.

[4] J. Lai, R. H. Deng, and Y. Li, "Expressive CP-ABE with partially hidden access structures," in *Proc. 7th ACM Symp. Inf., Comput. Commun. Secur. (ASIACCS)*, New York, NY, USA, 2012, pp. 18–19.

[5] Z. Wang and M. He, "CP-ABE with hidden policy from waters efficient construction," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 1, Jan. 2016, Art. no. 3257029, Accessed: Jan. 28, 2016, doi: 10.1155/2016/3257029.

[6] H. Cui, R. H. Deng, J. Lai, X. Yi, and S. Nepal, "An efficient and expressive ciphertext-policy attribute-based encryption scheme with partially hidden access structures, revisited," *Comput. Netw.*, vol. 133, pp. 157–165, Mar. 2018.

[7] L. Zhang, G. Hu, Y. Mu, and F. Rezaeibagha, "Hidden ciphertext policy attribute-based encryption with fast decryption for personal health record system," *IEEE Access*, vol. 7, pp. 33202–33213, Mar. 2019.

[8] P. Wayner, *Disappearing Cryptography: Information Hiding: Steganography & Watermarking*. San Francisco, CA, USA: Morgan Kaufmann, 2009.

[9] M. Dürmuth and D. M. Freeman, "Deniable encryption with negligible detection probability: An interactive construction," in *Proc. Eurocrypt*, Tallinn, Extonia, 2011, pp. 610–626.

[10] A. O'Neill, C. Peikert, and B. Waters, "Bi-deniable public-key encryption," in *Proc. Crypto*, Santa Barbara, CA, USA, 2011, pp. 525–542.

[11] M. Klonowski, P. Kubiak, and M. Kutylowski, "Practical deniable encryption," in *Proc. SOFSEM*, Nový Smokovec, Slovakia, 2008, pp. 599–609.

[12] P. Gasti, G. Ateniese, and M. Blanton, "Deniable cloud storage: Sharing files via public-key deniability," in *Proc. WPES*, Chicago, IL, USA, 2010, pp. 31–42.

[13] M. H. Ibrahim, "A method for obtaining deniable public-key encryption," *IJ Netw. Secur.*, vol. 8, no. 1, pp. 1–9, Apr. 2009.

[14] P.-W. Chi and C.-L. Lei, "Audit-free cloud storage via deniable attribute-based encryption," *IEEE Trans. Cloud Comput.*, vol. 6, no. 2, pp. 414–427, Apr. 2018.

[15] R. Canetti, S. Park, and O. Poburinnaya, "Fully bideniable interactive encryption," Cryptol. ePrint Arch., Tech. Rep., Dec. 2018. [Online]. Available: https://eprint.iacr.org/2018/1244

[16] C. Dwork, M. Naor, and A. Sahai, "Concurrent zero-knowledge," in *Proc. ACM STOC*, New York, NY, USA, 1998, pp. 409–418.

[17] M. Naor, "Deniable ring authentication," in *Proc. CRYPTO*, Santa Barbara, CA, USA, 2002, pp. 481–498.

[18] X. Deng, H. Zhu, and C. H. Lee, "Deniable authentication protocols," *IEE Proc. Comput. Digit. Techn.*, vol. 148, no. 2, pp. 101–104, Mar. 2001.

[19] L. Fan, C. X. Xu, and J. H. Li, "Deniable authentication protocol based on Deffie-Hellman algorithm," *Electron. Lett.*, vol. 38, no. 14, pp. 705–706, 2002.

[20] Z. Shao, "Efficient deniable authentication protocol based on generalized ElGamal signature scheme," *Comput. Standards Interfaces*, vol. 26, no. 5, pp. 449–454, Sep. 2004.

[21] D. Xiao, X. Liao, and K. Wong, "An efficient entire chaos-based scheme for deniable authentication," *Chaos, Solitons Fractals*, vol. 23, no. 4, pp. 1327–1331, Feb. 2005.

[22] M. Di Raimondo, R. Gennaro, and H. Krawczyk, "Deniable authentication and key exchange," in *Proc. 13th ACM Conf. Comput. Commun. Secur. (CCS)*, New York, NY, USA, 2006, pp. 400–409.

[23] M. Di Raimondo and R. Gennaro, "New approaches for deniable authentication," *J. Cryptol.*, vol. 22, no. 4, pp. 572–615, Oct. 2009.

[24] A. Fiat and M. Naor, "Broadcast encryption," in *Proc. CRYPTO*, Santa Barbara, CA, USA, 1993, pp. 480–491.

[25] J. Baek, R. S. Naini, and W. Susilo, "Efficient multi-receiver identity-based encryption and its application to broadcast encryption," in *Proc. PKC*, Les Diablerets, Switzerland, 2005, pp. 380–397.

[26] Y. Yu, B. Yang, X. Huang, and M. Zhang, "Efficient identity-based signcryption scheme for multiple receivers," in *Proc. ATC*, Hong Kong, 2007, pp. 13–21.

[27] B. Libert, K. G. Paterson, and E. A. Quaglia, "Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model," in *Proc. PKC*, Darmstadt, Germany, 2012, pp. 206–224.

[28] C.-I. Fan, L.-Y. Huang, and P.-H. Ho, "Anonymous multireceiver identity-based encryption," *IEEE Trans. Comput.*, vol. 59, no. 9, pp. 1239–1249, Sep. 2010.

[29] J. Zhang and P. Ou, "Privacy-preserving multi-receiver certificateless broadcast encryption scheme with de-duplication," *Sensors*, vol. 19, no. 15, p. 3370, Jul. 2019.

[30] L. Chen, J. Li, and Y. Zhang, "Adaptively secure anonymous identity-based broadcast encryption for data access control in cloud storage service," *KSII Trans. Internet Inf. Syst.*, vol. 13, no. 3, pp. 1523–1545, Mar. 2019.

[31] J. Li, Y. Wang, Y, Zhang, and J. Han, "Full verifiability for outsourced decryption in attribute based encryption," *IEEE Trans. Services Comput.*, to be published, doi: 10.1109/TSC.2017.2710190.

[32] J. Li, Q. Yu, and Y. Zhang, "Hierarchical attribute based encryption with continuous leakage-resilience," *Inf. Sci.*, vol. 484, pp. 113–134, May 2019.

[33] J. Li, W. Yao, Y. Zhang, H. Qian, and J. Han, "Flexible and fine-grained attribute-based data storage in cloud computing," *IEEE Trans. Services Comput.*, vol. 10, no. 5, pp. 785–796, Sep./Oct. 2017.

[34] J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen, "User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1767–1777, Jun. 2018.

[35] J. Li, Y. Zhang, J. Ning, X. Huang, G. S. Poh, and D. Wang, "Attribute based encryption with privacy protection and accountability for CloudIoT," *IEEE Trans. Cloud Comput.*, to be published, doi: 10.1109/TCC.2020.2975184.

[36] D. Apon, X. Fan, and F. H. Liu, "Deniable attribute based encryption for branching programs from LWE," in *Proc. TCC*, Beijing, China, 2016, pp. 299–329.

[37] S. Reddy, P. S. Reddy, and P. Sravanthi, "Audit free cloud storage via deniable attribute base encryption for protecting user privacy," *Int. J. Sci. Eng. Technol. Res.*, vol. 5, no. 17, pp. 3449–3451, Jul. 2016.

[38] D. Boneh, E. J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Proc. TCC*, Cambridge, MA, USA, 2005, pp. 325–341.

[39] H. Krawczyk and T. Rabin, "Chameleon signatures," in *Proc. NDSS*, San Diego, CA, USA, 2000.

[40] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Proc. PKC*, Taormina, Italy, 2011, pp. 53–70.

[41] A. Beimel, "Secure schemes for secret sharing and key distribution," Ph.D. dissertation, Dept. Comput. Sci., Israel Inst. Technol., Haifa, Israel, 1996.

[42] F. Zhang, R. S. Naini, and W. Susilo, "Id-based chameleon hashes from bilinear pairings," Cryptol. ePrint Arch., Tech. Rep., Apr. 2003. [Online]. Available: https://eprint.iacr.org/2003/208

[43] A. B. Lewko, "Tools for simulating features of composite order bilinear groups in the prime order setting," in *Proc. Eurocrypt*, Cambridge, U.K., 2012, pp. 318–335.

PO-WEN CHI received the B.S., M.S., and Ph.D. degrees in electrical engineering from National Taiwan University, in 2003, 2005, and 2016, respectively. From 2005 to 2016, he was an Engineer at the Institute for Information Industry, Taiwan. From 2016 to 2018, he was a Senior Engineer at Arcadyan Technology Corporation, Taiwan. He joined the Department of Computer Science and Information Engineering, National Taiwan Normal University, in 2018, as an Assistant Professor. His research interests include network security, applied cryptography, software-defined networking, and telecommunications.

MING-HUNG WANG received the B.S. degree in computer science and the M.S. degree in communication engineering from National Tsing-Hua University, in 2008 and 2010, respectively, and the Ph.D. degree from the Department of Electrical Engineering, National Taiwan University, in 2017. He joined the Department of Information Engineering and Computer Science, Feng Chia University, in 2018, as an Assistant Professor. His research interests include network security, social media analysis, and software-defined networking.

HUNG-JR SHIU received the B.S. and M.S. degrees in computer science and information engineering from National Chi Nan University, Puli, Taiwan, in 2004 and 2006, respectively, and the Ph.D. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 2018.

From 2018 to 2019, he was an Assistant Research Fellow with the Information and Communication Division, Cyber Warfare Technology Section, National Chung-San Institute of Science and Technology, Taoyuan, Taiwan. He joined the Department of Computer Science, Tunghai University, in 2020, as an Assistant Professor. His research interests include information security, signal processing, and algorithms.