White-Box Traceable CP-ABE for Cloud Storage Service: How to Catch People Leaking Their Access Credentials Effectively

Jianting Ning, Zhenfu Cao, *Senior Member, IEEE*, Xiaolei Dong, and Lifei Wei

Abstract—Ciphertext-policy attribute-based encryption (CP-ABE) has been proposed to enable fine-grained access control on encrypted data for cloud storage service. In the context of CP-ABE, since the decryption privilege is shared by multiple users who have the same attributes, it is difficult to identify the original key owner when given an exposed key. This leaves the malicious cloud users a chance to leak their access credentials to outsourced data in clouds for profits without the risk of being caught, which severely damages data security. To address this problem, we add the property of *traceability* to the conventional CP-ABE. To catch people leaking their access credentials to outsourced data in clouds for profits effectively, in this paper, we first propose two kinds of non-interactive commitments for traitor tracing. Then we present a fully secure traceable CP-ABE system for cloud storage service from the proposed commitment. Our proposed commitments for traitor tracing may be of independent interest, as they are both pairing-friendly and homomorphic. We also provide extensive experimental results to confirm the feasibility and efficiency of the proposed solution.

Index Terms—Ciphertext-policy attribute-based encryption, cloud storage, outsourced data security, white-box traceablity, commitment

1 Introduction

Emerging cloud computing replaces traditional outsourcing techniques and provides an efficient and cost-effective mechanism for organizations and individuals to enforce highly scalable and technology-enabled management on their data. As a new commercial and exciting paradigm, it has attracted much attention from both industrial and academic world. Cloud storage service enables cloud users to outsource their data to the cloud so that themselves or other authorized users can access the outsourced cloud data anywhere and anytime. Despite lots of benefits provided by cloud storage service, the concerns on data security are believed the main obstacles hindering the wide usage of cloud storage service. Cloud users may worry about the privacy of their outsourced data due to some unauthorized access to outsourced data (such as the loss of physical control of outsourced data, etc.). To address the data security concerns, encryption has been applied on the data before outsourcing. Meanwhile, in many cases, cloud users may want to share their outsourced data to some potential users without knowing who will receive it. Thus, a fine-grained access control over outsourced data is desired. Attribute-Based Encryption (ABE, [15]) is a highly promising approach to protect outsourced data and provide fine-grained access control for cloud storage service. In the context of CP-ABE, ciphertexts and keys are labeled with access policies and sets of descriptive attributes respectively. In a CP-ABE system for cloud storage service, data owners can specify access policies over attributes that the potential authorized cloud users should possess, and those cloud users who are authorized will be issued access credentials corresponding to their attribute sets and can get access to the outsourced data. A cloud user is authorized if the set of attributes he/she possesses satisfies the access policy specified by data owners. Intuitively, CP-ABE not only enables fine-grained access control over outsourced data in clouds, but also providing a reliable method to protect the outsourced data in clouds.

However, in practice, there exists an important issue that needs to be solved. In CP-ABE, a user who possesses a set of attributes could decrypt a ciphertext if his/her attributes pass through the ciphertext's access structure. The decryption privilege is shared by multiple users who have the same attributes but not associated with individuals. Accordingly, it is difficult to trace the malicious users who intentionally leak their keys. As a result, malicious users may deliberately leak their decryption keys to others for profits.

There are many examples in which key leakage undermines the intended purpose of the encryption system. As noted in [17], for example, Staddon et al. [40] present an ABE system for document redaction where parts of a document are 'blacked-out' by encrypting them. Those users whose attributes satisfying the redacted document's policy are then able to decrypt these blacked-out regions and fully read the content of the document. Legal documents might black-out the names of minors in a report for instance. Only law enforcement personnel, judges, or high ranking politicians are specified (by the policy) to be able to decrypt this

- *J. Ning is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: jtning@sjtu.edu.cn.*
- *Z. Cao and X. Dong are with Shanghai Key Lab for Trustworthy Computing, East China Normal University, Shanghai 200062, China. E-mail: {zfcao, dongxiaolei}@sei.ecnu.edu.cn.*
- *L. Wei is with the College of Information Technology, Shanghai Ocean University, Shanghai 201306, China. E-mail: Lfwei@shou.edu.cn.*

Manuscript received 17 Mar. 2016; revised 2 Sept. 2016; accepted 6 Sept. 2016. Date of publication 12 Sept. 2016; date of current version 29 Aug. 2018. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TDSC.2016.2608343

information. If a secret key is leaked to the press in such a scenario, the privacy of the minors in the document are violated and perhaps local laws too [17]. Another suggested usage for ABE, by Bethencourt et al. [3], is to encrypt documents by the FBI. The memos are encrypted by FBI agents so that only the people who have certain credentials are able to decrypt and access it. Since there is no recourse for delegating keys in the ABE system, a low paid agent might be tempted to sell his secret key to a private investigator or the press [17].

For cloud storage service, malicious data consumers may leak their access credentials to others for financial gain with almost no risk of getting caught. We take Alice (with attributes "$\{Alice, Department\ of\ Research, Engineer\}$") and Bob (with attributes "$\{Bob, Department\ of\ Research, Engineer\}$") as an example. As an expressive one-to-many encryption mechanism, CP-ABE enables data to be encrypted and outsourced in clouds under an access policy with role-based descriptive attributes, such as "$((Department\ of\ Research$ **AND** $Engineer)$ **OR** $Senior\ Engineer)$". Meanwhile, both Alice and Bob are able to delegate the decryption key (i.e., access credential) corresponding to attribute sets "$\{Department\ of\ Research, Engineer\}$". If one sells the key (or access credential) for such an attribute set on the Internet, it is hard to identity who is the original seller.

Functional Encryptions are usually used as prototypes for commercial applications. In some special cases of Functional Encryptions (such as Identity-Based Encryption, IBE), the decryption privilege is exclusive to the key owner and it is trivial to trace the owner of an exposed decryption key. Indeed, in all of one-to-one encryption mechanisms (such as IBE), the tracing task can be simply achieved by embedding the identity of a user into his/her secret key. However, in the case of CP-ABE (which is a one-to-many encryption mechanism), encrypted data is often shared with some potential users without knowing who will receive it. And simply embedding an identity into a user's secret key does not work, since this will conflict with the fine-grained access control property of CP-ABE. As a result, we need new ideas and techniques. Due to the nature of CP-ABE, those users who possess the same attributes share the same decryption privilege, and this gives them a chance to leak the secret key for commercial purpose without the risk of being caught. It is actually the *Malicious Key Delegation* issue. This malicious key delegation problem severely limits the practicability of CP-ABE. Therefore, it is necessary to consider the traceability of the malicious users who leak their decryption privilege in CP-ABE system. There are roughly two kinds of leaking the decryption privilege. The first one is to leak a decryption key, and the second one is to leak a decryption black-box. Correspondingly, there are roughly two kinds of traceability. The first one is *white-box traceability*, given a well-formed decryption key, a tracing algorithm can identify the malicious user who leaks this key. The second one is *black-box traceability*, given a decryption black-box, a tracing algorithm can identify the malicious user(s) who built the device using their secret key(s). As noted in [27], in the black-box traceability scenario, the decryption key and even the decryption algorithm of a given decryption black-box could be hidden. In this case, the tracing algorithm can still trace the malicious user(s) whose key(s) has been used in constructing the decryption black-box. Intuitively, black-box traceability is stronger than white-box traceability. While in practice, white-box traceability is more efficient and effective than black-box traceability, and is easier to implement. In this paper, we aim at achieving an efficient fully secure CP-ABE system for cloud storage service with white-box traceability, which resists malicious key delegation inherently and could trace malicious cloud consumers leaking their access credentials effectively.

1.1 Our Contribution

In this paper, we first propose two kinds of non-interactive commitments for traitor tracing: non-interactive traceable commitment with perfectly binding key and non-interactive traceable commitment with perfectly hiding key. We further present a fully secure white-box traceable CP-ABE system for cloud storage service from commitment with perfectly binding key (T-CPABE-pbCom system). Our system enables data owners to define access policies for their data to be outsourced, and any authorized cloud users will be issued access credentials corresponding to the sets of attributes they possess. For illegal leakage of access credential(s) to outsourced data, our system could trace the owner(s) of the leaked access credential(s).

To the best of our knowledge, this is the first fully secure white-box traceable CP-ABE system (for cloud storage service) from non-interactive traceable pairing-friendly commitment with perfectly binding key. We say a commitment is pairing-friendly if it is based on the Subgroup Decision assumption. In particular, this paper presents a new vision for traitor tracing in Functional Encryptions (CP-ABE as an example), that is, to achieve white-box traceability from commitment based on the Subgroup Decision assumption. We believe our technique of achieving white-box traceability may be a universal method for Functional Encryptions. Moreover, our new T-CPABE-pbCom system does not need to maintain an identity table T (which expands with the number of the users) as introduced in [28]. Instead, we adopt the commitment scheme to trace the malicious users, which is practical for cloud storage service. It is worth that, compared with [31], [32], [33], our new fully secure white-box traceable CP-ABE system removes the need of the identity table T without requiring any additional parameters.[1] This answers an interesting problem posed by [33]. To achieve the white-box traceability, we only need the group elements and the random secret parameters which have already been chosen and used in the underlying system. Our proposed commitments for traitor tracing may be of independent interest, as they are both homomorphic and easy to obtain.

We note that, in practice, the identity table T as used in [28] would grow linearly with the number of the users. Moreover, the table T will expand sharply in a large scale system, which causes a heavy burden for the storage space

1. Note that in [31], [33], they need to introduce the Shamir's $(\bar{t}, \bar{n})$ threshold scheme [39] and probabilistic encryption scheme (Enc, Dec) [13] to replace the role of the identity table T. Actually, they need to introduce some additional parameters (such as the $f(x)$ and $\bar{t}-1$ points for Shamir's $(\bar{t}, \bar{n})$ threshold scheme and $\bar{k}_1, \bar{k}_2$ for probabilistic encryption scheme (Enc, Dec)) to achieve the elimination of T. While in [32], they (also) need to introduce an extra Paillier-style encryption to achieve the elimination of T.

of T, especially for resource-constrained mobile devices. Also, with T expanding, the corresponding cost for traceability (i.e., searching K' in T used in [28]) and user revocation would become higher, which also causes an additional computation burden, especially for resource-constrained mobile devices. In practice, it is a trend to eliminate the identity table T (without requiring any additional parameters) in order to obtain an efficient and practical construction. It is easy to see that the property of removing the identity table without requiring any additional parameters widens our potential application ranges (especially for some resource-constrained applications), which makes our construction more practical.

1.2 Our Technique

In this section, we briefly introduce the main idea we utilize to realize fully secure white-box traceable CP-ABE system (for cloud storage service) from commitment. We give the full details in Sections 4 and 5.

Commitment schemes are basic ingredients in many cryptographic protocols. They are used to enable a party to commit to a value without revealing it. Later, the commitment is "opened", and it is guaranteed that the "opening" can yield only a single value determined in the committing phase [12]. Inspired by the two conflicting properties (i.e., binding and hiding) of commitment schemes, we utilize the commitment schemes to achieve the property of white-box traceability for CP-ABE. Specifically, when a user queries for his/her decryption key, the system will make a commitment to the user's identity and insert the commitment into his/her decryption key implicitly.

To explain this more precisely, we present two different kinds of pairing-friendly commitments for traitor tracing. The first one is a perfectly binding commitment, and actually, it is the cryptosystem from [5]. The second kind of commitment is a perfectly hiding one, and it is actually a Pedersen-style commitment [36]. To construct a white-box traceable CP-ABE, during the **`KeyGen`** algorithm, we first make a perfectly binding (or hiding) commitment to a user's identity, and let the commitment to be a necessary component for successful decryption. We then utilize the Boneh and Boyen's signature scheme to sign the commitment, and "inject" the signature into the user's decryption key. During the **`Trace`** algorithm, the system will use the perfectly binding (or hiding) key of the commitment to find out the identity of the malicious user. Before the tracing step, we take a *key sanity check* on the user's decryption key to check whether it is *well-formed* or not. Note that since the perfectly hiding commitment needs an additional table to record the openings of commitments which have made, in this paper, we only consider the construction of white-box traceable CP-ABE system from the perfectly binding commitment. In fact, one can also construct a white-box traceable CP-ABE system from the perfectly hiding commitment using the same method.

We adopt the construction and the proof methods from [24] in this work because of its efficiency and its elegant interface to fully security. Nevertheless, our system could also adapt to other constructions and proof methods that makes us believe our technique could be a universal method for other CP-ABE systems.

1.3 Related Work

Sahai et al. first introduced the notion of Fuzzy Identity-Based Encryption [38]. Later, Goyal et al. [15] formalized two notions of ABE (Ciphertext-policy ABE and Key-policy ABE). After that, many variants of ABE were proposed [3], [23], [24], [37]. However, these ABE constructions did not address the traceability problem. Li et al. [26] first proposed the notion of accountable CP-ABE to prevent illegal key sharing among colluding users. Later on, a multi-authority ciphertext-policy ABE scheme with accountability was proposed in [25], which allowed tracing the identity of a misbehavior user. But the ciphertext-policy in [26] and [25] can only be AND gate with wildcard, which is not expressive (i.e., supporting any monotone access structures). Katz et al. [18] introduced the notion of traceability in the context of predicate encryption. In [18], traceability was added to any inner-product predicate encryption with additional overhead linear in the number of the whole users. Later, Liu et al. [27] proposed an expressive CP-ABE with black-box traceability at the expense of sub-linear overhead. However, the black-box traceable systems in [18] and [27] have relatively huge public parameters and ciphertexts, which make them less practical. To prevent the key leakage in the black-box manner, Kiayias and Tang [19] addressed the problem of leakage deterring public key cryptography. In their systems, if a key owner leaks any partially working decryption box, some secret information that is embedded in the key owner's public key will be revealed to the recipient [19]. Later, Kiayias and Tang [20] generalized the leakage deterring public key cryptography in [19] from the single user setting to the multi-user setting. Though the techniques suggested in [19] and [20] are sound, they cannot be applied into the traceability in CP-ABE directly. To address the efficiency (compared with the black-box traceability), Liu et al. proposed a white-box [28] traceability CP-ABE system. It needs to maintain an identity table T to record the identities of all users for white-box traceability, which makes the system less practical. Ning et al. [31] proposed a practical large universe CP-ABE system supporting white-box traceability. Their system does not have to maintain the identity table, but they need to introduce additional parameters (i.e., the parameters for Shamir's threshold scheme and probabilistic encryption scheme) to remove the table. Ning et al. [32] later proposed an accountable authority CP-ABE with white-box traceability which also considered the misbehavior of the semi-trusted authority. For white-box traceability of malicious users, their system also needs to introduce an extra (independent) Paillier-style encryption to remove the identity table. Ning et al. [33] raised the above problem as an interesting problem to remove the the identity table T without introducing additional parameters.

1.4 Organization

Section 2 gives the formal definition and security model of fully secure white-box traceable CP-ABE system from commitment. Section 3 introduces the relevant background. Section 4 presents two non-interactive commitments for traitor tracing. Section 5 presents the construction of the T-CPABE-pbCom system as well as the security proofs. Section 6 gives the system performance and comparison between our work and some other related work. Some extensions of our

work are discussed in Section 7. Finally, Section 8 presents a brief conclusion and foresees our future work.

2 The Model of Fully Secure White-Box Traceable CP-ABE from Commitment

2.1 Definition

A Fully Secure White-Box Traceable CP-ABE system from Perfectly Binding Commitment (T-CPABE-pbCom system) is a fully secure CP-ABE system which can trace the malicious user by his/her decryption key through commitment. We enhance the original fully secure CP-ABE system by adding users' identities, commitments to these identities and a **Trace** algorithm to it. Specifically, following the notation of the CP-ABE system introduced in [24], [31], [32], [33], a T-CPABE-pbCom system consists of six algorithms as follows:

- $\mathbf{Setup}(1^\lambda, \mathcal{U}) \rightarrow (pk, msk)$: The algorithm takes as input a security parameter λ encoded in unary and the attribute universe description $\mathcal{U}$. It outputs the public parameters pk and the master secret key msk.
- $\mathbf{KeyGen}(pk, msk, id, S) \rightarrow sk_{id,S}$: The key generation algorithm takes as input the public parameters pk, the master secret key msk and a set of attributes S for a user with identity id. The algorithm makes a commitment to the user's identity id and inserts the commitment into the secret key $sk_{id,S}$ implicitly. It outputs a secret key $sk_{id,S}$ corresponding to S.
- $\mathbf{Encrypt}(pk, m, \mathbb{A}) \rightarrow \mathcal{T}$: The encryption algorithm takes as input the public parameters pk, a plaintext message m, and an access structure $\mathbb{A}$ over the universe of attributes. It outputs the ciphertext $\mathcal{T}$.
- $\mathbf{Decrypt}(pk, sk_{id,S}, \mathcal{T}) \rightarrow m$ or $\perp$: The decryption algorithm takes as input the public parameters pk, a secret key $sk_{id,S}$, and a ciphertext $\mathcal{T}$. If the set of attributes of the private key satisfies the access structure of the ciphertext, the algorithm outputs the plaintext m. Otherwise, it outputs $\perp$.
- $\mathbf{KeySanityCheck}(pk, sk_{id,S}) \rightarrow 1$ or 0 : The key sanity check algorithm takes as input the public parameters pk and a secret key $sk_{id,S}$. If $sk_{id,S}$ passes the key sanity check, the algorithm outputs 1. Otherwise, it outputs 0. The key sanity check is a deterministic algorithm [14], which is used to guarantee a secret key to be well-formed during the decryption process [33]. That is, the key sanity check algorithm is used to check whether a secret key can always be able to decrypt (the corresponding) ciphertexts.
- $\mathbf{Trace}(pk, msk, sk_{id,S}) \rightarrow id$ or $\intercal$: The tracing algorithm takes as input the public parameters pk, the master secret key msk and a secret key $sk_{id,S}$. The algorithm first runs $\mathbf{KeySanityCheck}(pk, sk_{id,S})$ and checks whether $sk_{id,S}$ is *well-formed* or not so as to determine whether the holder of $sk_{id,S}$ should be caught as a traitor. A secret key $sk_{id,S}$ is defined to be *well-formed* if $\mathbf{KeySanityCheck}(pk, sk_{id,S}) \rightarrow 1$. If $sk_{id,S}$ is well-formed, the system extracts the identity id from $sk_{id,S}$. More precisely, the system can extract the identity id from the commitment which is inserted in $sk_{id,S}$. It then outputs an identity id with which the $sk_{id,S}$ associates. Otherwise, it outputs a special symbol $\intercal$ indicates that the holder of $sk_{id,S}$ is honest.

2.2 Full Security

The security model of the T-CPABE-pbCom system is similar to that of the CP-ABE system in [24], excepting every key query is companied with an explicit identity. Below, we present the definition of full security for T-CPABE-pbCom system. It is parameterized by the security parameter $\lambda \in \mathbb{N}$ and is described by a security game between a challenger $\mathcal{C}$ and an attacker $\mathcal{A}$. The game proceeds as follows:

- *Setup* : $\mathcal{C}$ runs $\mathbf{Setup}(1^\lambda, \mathcal{U})$ and sends the public parameters pk to the $\mathcal{A}$.
- *Query Phase 1* : In this phase, $\mathcal{A}$ can adaptively query $\mathcal{C}$ for secret keys corresponding to sets of attributes $(id_1, S_1), (id_2, S_2), \ldots, (id_{Q_1}, S_{Q_1})$. For each (id_i, S_i), $\mathcal{C}$ calls $\mathbf{KeyGen}(pk, msk, id_i, S_i) \rightarrow sk_{id_i,S_i}$ and sends sk_{id_i,S_i} to $\mathcal{A}$.
- *Challenge* : $\mathcal{A}$ declares two equal length messages m_0, m_1 and an access structure $\mathbb{A}^*$. Note that this access structure cannot be satisfied by any of the queried attributes sets $(id_1, S_1), (id_2, S_2), \ldots, (id_{Q_1}, S_{Q_1})$. $\mathcal{C}$ flips a random coin $\beta \in \{0, 1\}$ and calls $\mathbf{Encrypt}(pk, m_\beta, \mathbb{A}^*) \rightarrow \mathcal{T}$. It sends $\mathcal{T}$ to $\mathcal{A}$.
- *Query Phase 2* : $\mathcal{A}$ adaptively queries $\mathcal{C}$ for the secret keys corresponding to sets of attributes $(id_{Q_1+1}, S_{Q_1+1}), \ldots, (id_Q, S_Q)$ with the restriction that none of these satisfies $\mathbb{A}^*$. For each (id_i, S_i), $\mathcal{C}$ calls $\mathbf{KeyGen}(pk, msk, id_i, S_i) \rightarrow sk_{id_i,S_i}$ and sends sk_{id_i,S_i} to $\mathcal{A}$.
- *Guess* : $\mathcal{A}$ outputs a guess $\beta' \in \{0, 1\}$ for β.

The advantage of an attacker in this game is defined to be $Adv = |Pr[\beta' = \beta] - 1/2|$.

Definition 1. *A traceable ciphertext-policy attribute-based encryption system from perfectly binding commitment is fully secure if all probabilistic polynomial-time (PPT) attackers have at most a negligible advantage in λ in the above security game.*

2.3 Traceability

We here give the traceability definition for the T-CPABE-pbCom system according to [32], [33]. It is described by a security game between a challenger $\mathcal{C}$ and an attacker $\mathcal{A}$. The game proceeds as follows:

- *Setup* : $\mathcal{C}$ calls $\mathbf{Setup}(1^\lambda, \mathcal{U})$ and sends the public parameters pk to $\mathcal{A}$.
- *Key Query* : $\mathcal{A}$ submits the sets of attributes $(id_1, S_1), \ldots, (id_q, S_q)$ to request the corresponding decryption keys. $\mathcal{C}$ calls $\mathbf{KeyGen}(pk, msk, id_i, S_i) \rightarrow sk_{id_i,S_i}$ and returns sk_{id_i,S_i} to $\mathcal{A}$.
- *Key Forgery* : $\mathcal{A}$ will output a decryption key sk_*. If $\mathbf{Trace}(pk, msk, sk_*) \neq$ and $\mathbf{Trace}(pk, msk, sk_*) \notin \{id_1, \ldots, id_q\}$, $\mathcal{A}$ wins the game. The advantage of $\mathcal{A}$ in this game is defined to be $\Pr[Trace(pk, msk, sk_*) \notin \{, id_1, \ldots, id_q\}]$.

Definition 2. *A traceable ciphertext-policy attribute-based encryption system from perfectly binding commitment is fully*

traceable if all PPT attackers have at most a negligible advantage in the above game.

2.4 Key Sanity Check

We give the definition of Key Sanity Check for the T-CPABE-pbCom system according to [1], [33]. In the subsequent security model, we formalize the intuition that a method should be provided to make sure that a user's decryption key issued by the system can always be able to decrypt ciphertexts for the user [1], [33]. It is described by the security game between a simulator and an attacker. On input a security parameter λ, a simulator invokes an attacker $\mathcal{A}$ on λ. The attacker $\mathcal{A}$ returns the public parameters pk, a ciphertext $\mathcal{T}$, and two distinct decryption keys $sk_{id,S}, sk'_{id,S}$ corresponding to the same set of attributes S and identity id. The attacker $\mathcal{A}$ wins the game if

(1) $\texttt{KeySanityCheck}(pk, sk_{id,S}) \rightarrow 1$.
(2) $\texttt{KeySanityCheck}(pk, sk'_{id,S}) \rightarrow 1$.
(3) $\texttt{Decrypt}(pk, sk_{id,S}, \mathcal{T}) \neq \perp$.
(4) $\texttt{Decrypt}(pk, sk'_{id,S}, \mathcal{T}) \neq \perp$.
(5) $\texttt{Decrypt}(pk, sk_{id,S}, \mathcal{T}) \neq \texttt{Decrypt}(pk, sk'_{id,S}, \mathcal{T})$.

We define $\Pr[\mathcal{A}\ wins]$ as the attacker $\mathcal{A}$'s advantage in the above game. It is easy to see that the intuition of "Key Sanity Check" is captured combining the related algorithms (i.e., **KeySanityCheck** and **Decrypt**) and the notion captured in the game described above.

3 Background

3.1 Notation

We define $[q] = \{1, 2, \ldots, q\}$ for $q \in \mathbb{N}$. By $\mathbb{Z}_p^{l \times n}$ we denote the set of matrices of size $l \times n$ with elements in $\mathbb{Z}_N$. The set of column vectors of length n (i.e., $\mathbb{Z}_N^{n \times 1}$) and the set of row vectors of length n (i.e., $\mathbb{Z}_N^{1 \times n}$) are the two special subsets. We denote by $x \xleftarrow{R} X$ the fact that x is picked uniformly at random from the finite set X.

3.2 Access Policy

Definition 3 (Access Structure [2], [33]). *Let S be the attribute universe. A collection (respectively, monotone collection) $\mathbb{A} \subseteq 2^S$ of non-empty sets of attributes is an access structure (respectively, monotone access structure) on S. A collection $\mathbb{A} \subseteq 2^S$ is called monotone if $\forall B, C \in \mathbb{A} : if\ B \in \mathbb{A}$ and $B \subseteq C$, then $C \in \mathbb{A}$. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.*

In CP-ABE, the role of the users is taken by the attributes. Thus, the access structure (respectively, monotone access structure) $\mathbb{A}$ will contain the authorized sets of attributes. If a user possesses an authorized set of attributes, then he/she can decrypt the corresponding ciphertext. Otherwise, the attribute set he/she possessed is unauthorized and he/she can't decrypt ciphertext. In this paper, we restrict our attention to the monotone access structure.

3.3 Linear Secret-Sharing Schemes

Definition 4 (Linear Secret-Sharing Schemes (LSSS) [2], [33]). *Let S denote the attribute universe and p denote a prime. A secret-sharing scheme $\prod$ with domain of secrets $\mathbb{Z}_p$ realizing access structure on S in called linear (over $\mathbb{Z}_p$) if: (1) The shares of a secret $s \in \mathbb{Z}_p$ for each attribute form a vector over $\mathbb{Z}_p$; (2) For each access structure $\mathbb{A}$ on S, there exists a matrix M with l rows and n columns called the share-generating matrix for $\prod$. For $i = 1, \ldots, l$, we define a function ρ labels row i of M with attribute $\rho(i)$ from the attribute universe S. When we consider the column vector $\vec{v} = (s, r_2, \ldots, r_n)$, where $s \in \mathbb{Z}_p$ is the secret to be shared and $r_2, \ldots, r_n \in \mathbb{Z}_p$ are randomly chosen. Then $M\vec{v} \in \mathbb{Z}_p^{l \times 1}$ is the vector of l shares of the secret s according to $\prod$. The share $(M\vec{v})_j$ "belongs" to attribute $\rho(j)$, where $j \in [l]$.*

As shown in [2], every linear secret-sharing scheme enjoys the linear reconstruction property, which defined as follows: we assume that $\prod$ is an LSSS for the access structure $\mathbb{A}$, $S' \in \mathbb{A}$ is an authorized set and let $I \subset \{1, 2, \ldots, l\}$ be defined as $I = \{i \in [l] \wedge \rho(i) \in S'\}$. There exist the constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that for any valid shares $\{\lambda_i = (M\vec{v})_i\}_{i \in I}$ of a secret s according to $\prod$, $\sum_{i \in I} \omega_i \lambda_i = s$. In additional, these constants $\{\omega_i\}_{i \in I}$ can be found in time polynomial in the size of the share-generating matrix M [2]. For any unauthorized set S'', no such $\{\omega_i\}$ exist.

3.4 Composite Order Bilinear Groups

Composite order bilinear groups are widely used in IBE and ABE systems, which are first introduced in [5]. We let $\mathcal{G}$ denote a group generator, which takes a security parameter λ as input and outputs a description of a bilinear group G. We define the output of $\mathcal{G}$ as $(p_1, p_2, p_3, G, G_T, e)$, where p_1, p_2, p_3 are distinct primes, G and G_T are cyclic groups of order $N = p_1 p_2 p_3$, and $e : G \times G \rightarrow G_T$ is a map such that: (1) Bilinearity: $\forall u, v \in G$ and $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$; (2) Non-degeneracy: $\exists g \in G$ such that $e(g, g)$ has order N in G_T.

We assume that group operations in G and G_T as well as the bilinear map e are computable in polynomial time with respect to λ. We refer to G as the *source group* and G_T as the *target group*, and assume the group descriptions of G and G_T include a generator of each group. Let G_{p_1}, G_{p_2}, and G_{p_3} be the subgroups of order p_1, p_2, and p_3 in G, respectively. Note that these subgroups are "orthogonal" to each other under the bilinear map e: for any $u_i \in G_{p_i}$ and $u_j \in G_{p_j}$ where $i \neq j$, $e(u_i, u_j) = 1$. Any element $E_N \in G$ can (uniquely) be expressed as $g_1^{r_1} g_2^{r_2} g_3^{r_3}$ for some values $r_1, r_2, r_3 \in \mathbb{Z}_N$, where g_1, g_2, g_3 are the generators of G_{p_1}, G_{p_2}, G_{p_3} respectively. And we will refer to $g_1^{r_1}, g_2^{r_2}, g_3^{r_3}$ as the "G_{p_1} part of E_N", "G_{p_2} part of E_N" and "G_{p_3} part of E_N", respectively. Assume $G_{p_1 p_2}$ be the subgroups of order $p_1 p_2$ in G. Similarly, any element $E_{p_1 p_2} \in G p_1 p_2$ can be expressed as the product of an element from G_{p_1} and an element from G_{p_2}.

3.5 Complexity Assumptions

In this section, we will present some complexity assumptions according to [24]. Consider groups G with orders $N = p_1 p_2 p_3$. For any non-empty set $Z \subseteq \{1, 2, 3\}$, there is a corresponding subgroup of G of order $\prod_{i \in Z} p_i$. And let this subgroup be G_Z.

Assumption 1. *[24] Given a group generator $\mathcal{G}$, define the following distribution:*

$$\mathbb{G} = (N = p_1p_2p_3, G, G_T, e) \xleftarrow{R} \mathcal{G}, \alpha, s \xleftarrow{R} \mathbb{Z}_N,$$
$$g_1 \xleftarrow{R} G_{p_1}, g_2, X_2, Y_2 \xleftarrow{R} G_{p_2}, g_3 \xleftarrow{R} G_{p_3}, D = (\mathbb{G}, g_1,$$
$$g_2, g_3, g_1^{\alpha}X_2, g_1^{s}Y_2), T_0 = e(g_1, g_1)^{\alpha s}, T_1 \xleftarrow{R} G_T.$$

An algorithm $\mathcal{A}$'s advantage in breaking this assumption is: $Adv^1_{\mathcal{G},\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(D, T_0) = 1] - \Pr[\mathcal{A}(D, T_1) = 1]|$. We say that $\mathcal{G}$ satisfies Assumption 1 if $Adv^1_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ for any polynomial time algorithm $\mathcal{A}$.

Assumption 2. *(The General Subgroup Decision Assumption): [24] Given a group generator $\mathcal{G}$ and a collection of non-empty subsets of $\{1, 2, 3\}$ $Z_0, Z_1, \ldots, Z_k$ where each Z_i for $i \geq 2$ satisfies either $Z_0 \cap Z_i = \phi = Z_1 \cap Z_i$ or $Z_0 \cap Z_i \neq \phi \neq Z_1 \cap Z_i$. Define the following distribution:*

$$\mathbb{G} = (N = p_1p_2p_3, G, G_T, e) \xleftarrow{R} \mathcal{G}, g_{Z_2} \xleftarrow{R} G_{Z_2}, \ldots,$$
$$g_{Z_k} \xleftarrow{R} G_{Z_k}, D = (\mathbb{G}, g_{Z_2}, \ldots, g_{Z_k}), T_0 \xleftarrow{R} G_{Z_0}, T_1 \xleftarrow{R} G_{Z_1}.$$

Fixing the collection of sets $Z_0, Z_1, \ldots, Z_k$, the advantage of an algorithm $\mathcal{A}$ in breaking this assumption is: $Adv^{SD}_{\mathcal{G},\mathcal{A}}(\lambda) = |Pr[\mathcal{A}(D, T_0) = 1] - Pr[\mathcal{A}(D, T_1) = 1]|$. We say that $\mathcal{G}$ satisfies the General Subgroup Decision Assumption if $Adv^{SD}_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ for any polynomial time algorithm $\mathcal{A}$ and any suitable collection of subsets $Z_0, Z_1, \ldots, Z_k$. Note that parameterized by the choice of the sets $Z_0, Z_1, \ldots, Z_k$, this assumption can be thought of as a family of assumptions.

Assumption 3. *(The Three Party Diffie-Hellman Assumption in a Subgroup): [24] Given a group generator $\mathcal{G}$, define the following distribution:*

$$\mathbb{G} = (N = p_1p_2p_3, G, G_T, e) \xleftarrow{R} \mathcal{G}, x, y, z \xleftarrow{R} \mathbb{Z}_N,$$
$$g_1 \xleftarrow{R} G_{p_1}, g_2 \xleftarrow{R} G_{p_2}, g_3 \xleftarrow{R} G_{p_3}, D = (\mathbb{G}, g_1,$$
$$g_2, g_3, g_2^{x}, g_2^{y}, g_2^{z}), T_0 = g_2^{xyz}, T_1 \xleftarrow{R} G_{p_2}.$$

An algorithm $\mathcal{A}$'s advantage in breaking this assumption is: $Adv^{3DH}_{\mathcal{G},\mathcal{A}}(\lambda) = |Pr[\mathcal{A}(D, T_0) = 1] - Pr[\mathcal{A}(D, T_1) = 1]|$. We say that $\mathcal{G}$ satisfies the Three Party Diffie-Hellman Assumption in a Subgroup if $Adv^{3DH}_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ for any polynomial time algorithm $\mathcal{A}$.

Assumption 4. *(The Source Group q-Parallel BDHE Assumption in a Subgroup): [24] Given a group generator $\mathcal{G}$ and a positive integer q, define the following distribution:*

$$\mathbb{G} = (N = p_1p_2p_3, G, G_T, e) \xleftarrow{R} \mathcal{G}, c, d, f, b_1, \ldots, b_q \xleftarrow{R} \mathbb{Z}_N,$$
$$g_1 \xleftarrow{R} G_{p_1}, g_2 \xleftarrow{R} G_{p_2}, g_3 \xleftarrow{R} G_{p_3},$$
$$D = (\mathbb{G}, g_1, g_2, g_3, g_2^{f}, g_2^{df}, g_2^{c}, g_2^{c^2}, \ldots, g_2^{c^q}, g_2^{c^{q+2}}, \ldots, g_2^{c^{2q}},$$
$$g_2^{c^i/b_j}\ \forall i \in [2q]\backslash\{q+1\}, j \in [q],$$
$$g_2^{dfb_j}\ \forall j \in [q], g_2^{dfc^ib_{j'}/b_j}\ \forall i \in [q], j, j' \in [q]\ s.t.\ j \neq j'),$$
$$T_0 = g_2^{dc^{q+1}}, T_1 \xleftarrow{R} G_{p_2}.$$

An algorithm $\mathcal{A}$'s advantage in breaking this assumption is: $Adv^{q}_{\mathcal{G},\mathcal{A}}(\lambda) = |Pr[\mathcal{A}(D, T_0) = 1] - Pr[\mathcal{A}(D, T_1) = 1]|$. We say that $\mathcal{G}$ satisfies the Source Group q-Parallel BDHE Assumption in a Subgroup if $Adv^{q}_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ for any polynomial time algorithm $\mathcal{A}$.

Assumption 5. *(q-SDH assumption [4]) : Let G be a bilinear group of prime order p and g be a generator of G, the q-Strong Diffie-Hellman (q-SDH) problem in G is defined as follows: given a $(q+1)$-tuple $(g, g^x, g^{x^2}, \ldots, g^{x^q})$ as input, output a pair $(c, g^{1/(c+x)}) \in \mathbb{Z}_p \times G$. An algorithm $\mathcal{A}$ has advantage ϵ in solving q-SDH in G if $\Pr[\mathcal{A}(g, g^x, g^{x^2}, \ldots, g^{x^q}) = (c, g^{1/(c+x)})] \geq \epsilon$, where the probability is over the random choice of x in $\mathbb{Z}_p^*$ and the random bits consumed by $\mathcal{A}$.*

We say that the (q, t, ϵ)-SDH assumption holds in G if no t-time algorithm has advantage at least ϵ in solving the q-SDH problem in G.

3.6 Commitment Schemes

We give the definition of commitment schemes according to [12]. A commitment scheme is a two-phase interactive protocol between two parties, called the sender S and the receiver R. The sender S can commit itself to a random value r such that the following two conflicting requirements are satisfied.

1. Hiding: At the end of the first phase (called the commitment phase), R does not gain any knowledge of S's value. This requirement must be satisfied even if the receiver tries to cheat.
2. Binding: Given the transcript of the interaction in the commitment phase, there exists at most one value that R can later (i.e., in the second phase, also called the reveal phase) accept as a legal "opening" of the commitment. This requirement must also be satisfied even if the sender tries to cheat.

In addition, one should require that the protocol be viable, in the sense that if both S and R honestly follow it, then at the end of the reveal phase R gets the value committed by S.

4 Non-Interactive Commitments for Traitor Tracing

In this section, we present commitment schemes based on the Subgroup Decision Assumption for traitor tracing according to [5], [16], [36].

4.1 Non-Interactive Traceable Commitment with Perfectly Binding Key

In this section, we present a traceable non-interactive commitment with perfectly binding key for traitor tracing.

- **`Perfectly Binding Key Generation`**$(1^\lambda) \rightarrow (pk, ek)$: The algorithm runs the group generator algorithm $\mathfrak{g}(1^\lambda)$ and gets the groups and the bilinear mapping description $GD = (p, q, g, G, G_T, e)$, where p, q are primes, (G, G_T) are groups of order $n = pq$, e is the bilinear mapping and g is a random generator of G. The algorithm randomly chooses $x \in \mathbb{Z}_q^*$, and computes $h = g^{xp}$. Then, it sets (G, G_T, e, g, n, h) as the public parameters pk and q as the perfectly binding key ek.

- **Commitment**$(pk, id) \rightarrow c$: The algorithm takes as input the public parameters pk and the identity of a user id.[2] To commit to the identity id, the algorithm chooses a random value $r \in \mathbb{Z}_n$ and computes $com_{pb}(id, r) = c = g^{id}h^r$. It then outputs c.
- **Trace**$(ek, c) \rightarrow id$: The algorithm takes as input the perfectly binding key ek and a commitment c. The algorithm computes $c^q = (g^{id}h^r)^q = (g^q)^{id}$, and exhaustively searches for id. Then it outputs the identity id.

Note that the traceable commit scheme described above is actually a variation of the cryptosystem from [5]. And it is an extractable commitment. As noted in [5], the commit scheme described above is homomorphic. Specifically, for all identities and randomizers we have $com_{pb}(id + id', r + r') = g^{id+id'}h^{r+r'} = g^{id}h^r g^{id'}h^{r'} = com_{pb}(id, r)com_{pb}(id', r')$. We insert the identity of the user into the extractable commitment with perfectly binding key implicitly, and use the perfectly binding key (i.e., the extractable key) to trace the traitor. As noted in [5], decryption in this scheme takes polynomial time in the size of the identity space, therefore, the scheme as described above can only be used to encrypt short messages. Clearly one can use the commitment to commit longer identifies, such as session keys, using any mode of operation that converts a cipher on a short block into a cipher on an arbitrary long block. And it is noted that one can speed-up decryption by precomputing a (polynomial-size) table of powers of g^q so that decryption can occur in constant time [5].

4.2 Non-Interactive Traceable Commitment with Perfectly Hiding Key

In this section, we present a traceable non-interactive commitment with perfectly hiding key for traitor tracing.

- **Perfectly hiding Key Generation**$(1^\lambda) \rightarrow (pk, ek)$: The algorithm runs the group generator algorithm $\mathfrak{g}(1^\lambda)$ and gets the groups and the bilinear mapping description $GD = (p, q, g, G, G_T, e)$, where p, q are primes, (G, G_T) are groups of order $n = pq$, e is the bilinear mapping and g is a random generator of G. The algorithm randomly chooses $x \in \mathbb{Z}_n^*$, and computes $h = g^x$. Then, it sets (G, G_T, e, g, n, h) as the public parameters pk and x as the perfectly hiding key ek. The algorithm initializes an empty table Ta, which is used to store the openings of commitments.
- **Commitment**$(pk, id) \rightarrow c$: The algorithm takes as input the public parameters pk and the identity of the traitor id. To commit to the message $id \in \mathbb{Z}_p$, the algorithm chooses a random value $r \in \mathbb{Z}_n$ and computes the commitment $com_{ph}(id, r) = c = g^{id}h^r$. Then it outputs c. Finally, the algorithm puts the tuple (r, id) into the table Ta.
- **Trace**$(ek, c, Ta) \rightarrow id$: The algorithm takes as input the perfectly hiding key ek, a commitment c and a table Ta. The algorithm searches the opening (r, id) of the commitment c in the table, and outputs the identity of the traitor id from the opening (r, id).

2. We assume the identity space consists of integers in the set $\{0, 1, \ldots, I\}$ with $I < p$.

Note that the commit scheme described above is actually a variation of the standard Pedersen commitment [36]. It is perfectly hiding since the commitment is uniformly distributed in G no matter what the value id is. And even with unlimited computational power, it is impossible for an attacker to learn any information about the value id from c, because the commitments of any two numbers $id, id' \in \mathbb{Z}_p$ have exactly the same distribution. As the Pedersen commitment [36], the commit scheme described above is homomorphic. Specifically, for all identities and randomizers we have $com_{ph}(id + id', r + r') = g^{id+id'}h^{r+r'} = g^{id}h^r g^{id'}h^{r'} = com_{ph}(id, r)$ $com_{ph}(id', r')$. We insert the identity of the user into the commitment with perfectly hiding key implicitly, and adopt the opening table to store the openings of the commitments to trace traitors. In practice, for each user who asks for a secret key, the traitor tracing system makes a commitment with perfectly hiding key to the user's identity and insert the commitment into the user's secret key implicitly. Meanwhile, for each user, the traitor tracing system employs an opening table to store the opening of the commitment corresponding to the user's secret key. Namely, the traitor tracing system keeps a record of openings of all users who have asked for their secret keys. When a leaked secret key is found in the wild or sold on the Internet, since the system only knows the openings of all users, a trace algorithm is needed to find out who leaks the secret key. Specifically, the system gets the leaked secret key and runs the trace operation. The trace operation first obtains (or extracts) the (implicit) commitment part of the leaked secret key, it then searches the opening (r, id) of the commitment in the opening table, and outputs the identity of the traitor id from the opening (r, id).

5 Our T-CPABE-pbCom System

5.1 Construction

In this section, we propose the construction of our new fully secure white-box traceable CP-ABE from commitment with perfectly binding key proposed in Section 4.1, which is as expressive and secure as the CP-ABE scheme in [24]. We adopt the pairing-friendly commitment with perfectly binding key to obtain the white-box traceability. In the following, based on the construction in [24], we present the concrete construction of our new system.

- **Setup**$(1^\lambda, \mathcal{U}) \rightarrow (pk, msk)$: The algorithm calls the group generator $\mathcal{G}$ with 1^λ as input and gets a bilinear group G of order $N = p_1p_2p_3$ (three distinct primes, whose size is determined by 1^λ), G_{p_i} the subgroup of order p_i in G, and g_3 the generator of the subgroup G_{p_3}. Then the algorithm chooses exponents $\alpha, a, b, \kappa \in \mathbb{Z}_N$ randomly. It also chooses group elements $g \in G_{p_1}$ and $R_0 \in G_{p_3}$ randomly. It then chooses a random value $u_i \in \mathbb{Z}_N$ for each attribute $i \in \mathcal{U}$. In addition, it chooses an efficiently computable injective encoding $INJ : G_N \rightarrow \mathbb{Z}_N$.[3] The public parameters are set to $pk = (N, g, g^a, g^b, g^\kappa, e(g, g)^\alpha,$

3. Actually, an almost injective encoding is sufficient for our purpose. Most interesting groups allow for such an encoding [7], [9], [11]. If such an encoding is not available one can also use a target collision resistant hash function $TCR : G_N \rightarrow \mathbb{Z}_N$, see [21] for more details.

$\{\mathcal{U}_i = g^{u_i}\}_{i\in\mathcal{U}}$). The master secret key is set to $msk = (p_1, a, g^{\alpha}, \kappa, g_3, R_0)$.

- **KeyGen**$(pk, msk, id, S) \rightarrow sk_{id,S}$ [4]: The algorithm randomly chooses $c, t, r \in \mathbb{Z}_N$, $h \in G_{p_1}$, $R, R', R'', R'_0, R''_0 \in G_{p_3}$, and $R_i \in G_{p_3}$ for each $i \in S$. The secret key $sk_{id,S}$ is set as follows:

$$\langle K = g^{\frac{\alpha}{a+T}} g^{btr} g^{c\kappa} R,\ K' = g^c h^r R',\ K'' = g^{tr} R'',$$

$$L = R_0^{id} h^{\kappa r},\ L' = g^{ac} h^{ar} R'_0,\ L'' = g^{atr} R''_0,$$

$$T = INJ(L),\ \{K_i = \mathcal{U}_i^{(a+T)tr} R_i\}_{i\in S},\ S\rangle.$$

- **Encrypt**$(pk, m, (A, \rho)) \rightarrow \mathcal{T}$: The algorithm takes as input the public parameters pk, a plaintext message m, and the access structure encoded in an LSSS policy (where A is an $l \times n$ matrix and ρ is a map from each row A_j of A to an attribute $\rho(j)$). Then it randomly chooses $\overrightarrow{y} = (s, y_2, \ldots, y_n)^{\perp} \in \mathbb{Z}_N^{n\times 1}$, where s is the random secret to be shared among the shares according to Section 3.3. And it randomly chooses $r_j \in \mathbb{Z}_N$ for each row A_j of A. The ciphertext $\mathcal{T}$ is set as follows:

$$\langle (A, \rho), C = m \cdot e(g, g)^{\alpha s}, C_0 = g^s, C_1 = (g^a)^s,$$

$$C_2 = (g^{\kappa})^s, \{C_{j,1} = (g^b)^{A_j \overrightarrow{y}} \mathcal{U}_{\rho(j)}^{-r_j}, C_{j,2} = g^{r_j}\}_{j\in[l]}\rangle.$$

- **Decrypt**$(pk, sk_{id,S}, \mathcal{T}) \rightarrow m$ or $\perp$: It first parses the $sk_{id,S}$ to $(K, K', K'', L, L', L'', T, \{K_i\}_{i\in S}, S)$ and $\mathcal{T}$ to $((A, \rho), C, C_0, C_1, C_2, \{C_{j,1}, C_{j,2}\}_{j\in[l]})$. If the attribute set S cannot satisfy the access structure (A, ρ) of $\mathcal{T}$ (i.e., S is not an authorized set of the access policy), the algorithm outputs $\perp$. Otherwise, the algorithm computes constants $\omega_j \in \mathbb{Z}_N$ such that $\sum_{\rho(j)\in S} \omega_j A_j = (1, 0, \ldots, 0)$. Then it computes

$$E = \Pi_{\rho(j)\in S}(e(C_{j,1}, (K'')^T L'') e(C_{j,2}, K_{\rho(j)}))^{\omega_j},$$

$$F = e((C_0)^T C_1, KL) e(C_2, (K')^T L')^{-1},$$

$$F/E = e(g, g)^{\alpha s}.$$

Then m can be recovered as $C/(F/E)$.

Correctness.

For each j,

$$e(C_{j,1}, (K'')^T L'') e(C_{j,2}, K_{\rho(j)}) = e(g, g)^{btr(a+T)A_j \overrightarrow{y}},$$

so we have

$$E = e(g, g)^{btrs(a+T)}.$$

$$F = e((C_0)^T C_1, KL) e(C_2, (K')^T L')^{-1} = e(g, g)^{\alpha s} \cdot e(g, g)^{sbtr(a+T)}.$$

4. We note that the identity space of the input id consists of integers in the set $\{0, 1, \ldots, I\}$ with I to be polynomial in λ such that $I << p_3$. Clearly one can use the commitment to commit longer identifies, such as session keys, using any mode of operation that converts a cipher on a short block into a cipher on an arbitrary long block [5].

- **KeySanityCheck**$(pk, sk) \rightarrow 1$ or 0 : The algorithm takes as input the public parameters pk and a secret key sk. sk passes the key sanity check if
  (1) The key is in the form of $(K, K', K'', L, L', L'', T, \{K_i\}_{i\in S})$, and $K, K', K'', L, L', L'', \{K_i\}_{i\in S} \in G, T \in Z_N$.
  (2) $e(L', g) = e(K', g^a) \neq 1$.
  (3) $e(L'', g) = e(K'', g^a) \neq 1$.
  (4) $e(KL, g^a \cdot g^T) = e(g, g)^{\alpha} e((K')^T L', g^{\kappa}) \cdot e((K'')^T L'', g^b) \neq 1$.
  (5) $\exists i \in S,\ s.t.\ e(K_i, g) = e(\mathcal{U}_i, (K'')^T L'') \neq 1$.
  (6) $T = INJ(L)$.

If sk passes the key sanity check, the algorithm outputs 1. Otherwise, it outputs 0.

- **Trace**$(pk, msk, sk) \rightarrow id$ or $\intercal$: If **KeySanityCheck**$(pk, sk) \rightarrow 0$, the algorithm outputs $\intercal$. Otherwise, sk is a well-formed decryption key, and the algorithm will use the perfect binding key (which is inserted in the msk implicitly) to extract id from $L = R_0^{id} h^{\kappa r}$ in sk as follows:
  (1) The algorithm computes $L^{p_1} = (R_0^{id} h^{\kappa r})^{p_1} = (R_0^{p_1})^{id}$ and let $\hat{R}_0 = R_0^{p_1}$.
  (2) Then it computes the discrete log (i.e., the identity id) of $(R_0^{p_1})^{id}$ to identify the malicious user. To recover id, it suffices to compute the discrete log of $(R_0^{p_1})^{id}$ base $\hat{R}_0$. Note that since the system knows the secrets R_0, p_1 and the identity $id \in \{0, 1, \ldots, I\}$ with I to be polynomial in λ such that $I << p_3$, the system can exhaustively search for id alternatively, use Pollard's lambda method [5], [30] to recover id.

5.2 Full Security Proof

In the full security proof, although we can give a proof which is directly based on the Assumption 1, the general subgroup decision assumption, the three party diffie-hellman assumption in a subgroup, and the source group q-Parallel BDHE assumption in a subgroup as [24] does, for simplicity, we will reduce the full security of our proposed system to that of Lewko and Waters's system in [24] which is proved fully secure under the same assumptions. By S_{cpabe}, S_{ccpabe} we denote the CP-ABE system in [24] and our system, respectively. Note that the security model of our system S_{ccpabe} is almost same with that of the system S_{cpabe} in [24], excepting every key query is companied with an explicit identity.

Lemma 1. *[24] If Assumption 1, the general subgroup decision assumption, the three party diffie-hellman assumption in a subgroup, and the source group q-Parallel BDHE assumption in a subgroup hold, then S_{cpabe} is fully secure.*

Full security of our proposed system S_{ccpabe}:

Lemma 2. *If S_{cpabe} is fully secure in the security game of [24], then S_{ccpabe} is fully secure in the security game of Section 2.2.*

Proof. Suppose there exists a PPT attacker $\mathcal{A}$ that has advantage $Adv_{\mathcal{A}} S_{ccpabe}$ in adaptively breaking our proposed system S_{ccpabe}. We construct a PPT algorithm $\mathcal{B}$ that has advantage $Adv_{\mathcal{B}} S_{cpabe}$ in adaptively breaking the

underlying CP-ABE system S_{cpabe}, which equals to $Adv_{\mathcal{A}}S_{ccpabe}$.

- *Setup* : S_{cpabe} gives $\mathcal{B}$ the public parameters $pks_{cpabe} = (N, g, g_3, g^b, g^\kappa, e(g,g)^\alpha, \{\mathcal{U} = g^{u_i}\}_{i\in\mathcal{U}})$.[5] $\mathcal{B}$ chooses $a \in \mathbb{Z}_N$, $R_0 \in G_{p_3}$ randomly. Let $INJ : G_N \to \mathbb{Z}_N$ be an efficiently computable injective encoding. $\mathcal{B}$ gives the public parameters $pk = (N, g, g^a, g^b, g^\kappa, e(g,g)^\alpha, \{\mathcal{U}_i = g^{u_i}\}_{i\in\mathcal{U}})$ to $\mathcal{A}$.
- *Query Phase 1* : The attacker $\mathcal{A}$ will submit (id, S) to $\mathcal{B}$ to query a decryption key, then $\mathcal{B}$ submits S to S_{cpabe} and gets the corresponding decryption key as follows: $\langle \tilde{K} = g^\alpha g^{b\tilde{t}} g^{\tilde{c}\kappa} R,\ \tilde{K}' = g^{\tilde{c}} R',\ \tilde{K}'' = g^{\tilde{t}} R'',\ \{\tilde{K}_i = \mathcal{U}_i^{\tilde{t}} R_i\}_{i\in S},\ S\rangle$. $\mathcal{B}$ randomly chooses r, $h' \in \mathbb{Z}_N$, $R_0', R_0'' \in G_{p_3}$, and computes $h = g^{h'} \in G_{p_1}$, $L = R_0^{id}(g^\kappa)^{h'r} = R_0^{id} h^{\kappa r}, T = INJ(L)$. It sets $tr = \frac{\tilde{t}}{a+T}, c = \frac{\tilde{c}}{a+T}$ implicitly, then computes

$$
\begin{aligned}
K &= (\tilde{K})^{\frac{1}{a+T}} = (g^\alpha g^{b\tilde{t}} g^{\tilde{c}\kappa})^{\frac{1}{a+T}} \\
&= g^{\frac{\alpha}{a+T}} g^{btr} g^{ck} R^{\frac{1}{a+T}}, \\
K' &= (\tilde{K}')^{\frac{1}{a+T}} h^r = (g^{\tilde{c}} R')^{\frac{1}{a+T}} h^r = g^c h^r R'^{\frac{1}{a+T}}, \\
K'' &= (\tilde{K}'')^{\frac{1}{a+T}} = (g^{\tilde{t}} R'')^{\frac{1}{a+T}} = g^{tr} R''^{\frac{1}{a+T}}, \\
L' &= (\tilde{K}')^{\frac{a}{a+T}} h^{ar} = (g^{\tilde{c}} R')^{\frac{a}{a+T}} h^r \\
&= g^{ac} h^{ar} R'^{\frac{a}{a+T}} R_0', \\
L'' &= (\tilde{K}'')^{\frac{a}{a+T}} = (g^{\tilde{t}} R'')^{\frac{a}{a+T}} = g^{atr} R''^{\frac{a}{a+T}} R_0'', \\
\{K_i &= \tilde{K}_i = \mathcal{U}_i^{\tilde{t}} R_i = \mathcal{U}_i^{(a+T)tr} R_i\}_{i\in S}.
\end{aligned}
$$

  $\mathcal{B}$ sends the decryption key $sk_{id,S} = \langle K, K', K'', L, L', L'', T, \{K_i\}_{i\in S}\rangle$ to $\mathcal{A}$.
- *Challenge* : The attacker $\mathcal{A}$ outputs two equal length messages (m_0, m_1) and an LSSS matrix (A^*, ρ), and sends them to $\mathcal{B}$. $\mathcal{B}$ submits $((A^*, \rho), m_0, m_1)$ to S_{cpabe}, and gets the challenge ciphertext as follows: $\langle (A^*, \rho), \tilde{C} = m_\beta \cdot e(g,g)^{\alpha s},\ \tilde{C}_0 = g^s, \tilde{C}_1 = (g^\kappa)^s, \{\tilde{C}_{j,1} = (g^b)^{A_j \overrightarrow{y}} \mathcal{U}_{\rho(j)}^{-r_j}, \tilde{C}_{j,2} = g^{r_j}\}_{j\in[l]}\rangle$. $\mathcal{B}$ sets $C = \tilde{C}, C_0 = \tilde{C}_0, C_1 = (\tilde{C}_0)^a = g^{as}, C_2 = \tilde{C}_1, C_{j,1} = \tilde{C}_{j,1}, C_{j,2} = \tilde{C}_{j,2}$. Finally, $\mathcal{B}$ gives the challenge ciphertext $\mathcal{T} = \langle (A^*, \rho), C, C_0, C_1, C_2,\ \{C_{j,1}, C_{j,2}\}_{j\in[l]}\rangle$ to $\mathcal{A}$.
- *Query Phase 2* : This phase is the same with Phase 1.
- *Guess* : The attacker outputs his guess β', and gives it to $\mathcal{B}$. Then $\mathcal{B}$ gives β' to S_{cpabe}.

Since the distributions of the public parameters, decryption keys and challenge ciphertext in the above game are the same as that in the real system, we have $Adv_{\mathcal{B}}S_{cpabe}$=$Adv_{\mathcal{A}}S_{ccpabe}$. □

Theorem 1. *If Assumption 1, the general subgroup decision assumption, the three party diffie-hellman assumption in a subgroup, and the source group q-Parallel BDHE assumption in a subgroup hold, then our proposed system is fully secure.*

Proof. It follows directly from Lemmas 1 and 2. □

5. Note that g_3 is a generator of G_{p_3} which is the master secret key in [24], here we move it to the public parameters.

5.3 Traceability Proof

We use a proof method from [4], [28], [31], [33].

Theorem 2. *If the subgroup decision assumption and the q-SDH assumption hold, our T-CPABE-pbCom system is fully traceable.*

Proof. Suppose there is a PPT attacker $\mathcal{A}$ achieving a non-negligible advantage ϵ in winning the traceability game after making q' key queries, we construct a PPT algorithm $\mathcal{B}$ that can break the subgroup decision assumption with sets $Z_0 := \{1,3\}, Z_1 := \{1,2,3\}, Z_2 := \{1\}, Z_3 := \{3\}, Z_4 := \{1,2\}, Z_5 := \{2,3\}$ or q-SDH assumption (without loss of generality, assuming $q = q' + 1$) with non-negligible advantage.

$\mathcal{B}$ is given an instance of q-SDH problem and an instance of the subgroup decision assumption problem as follows.

- $\mathcal{B}$ is given an instance of q-SDH problem $INS_{SDH} = (N, G, G_T, e, p_1, p_2, p_3, g_1, g_1^a, \ldots, g_1^{a^q})$, where G is a bilinear group of order $N = p_1p_2p_3$ (p_1, p_2, p_3 are three distinct primes), G_{p_i} is the subgroup of order p_i in G (where $1 \le i \le 3$), $e : G \times G \to G_T$ is a bilinear map, $g_1 \in G_{p_1}$ and $a \in \mathbb{Z}_{p_1}^*$ and $(g_1, g_1^a, \ldots, g_1^{a^q})$ is an instance of q-SDH problem in the subgroup G_{p_1}. Note that $\mathcal{B}$ is given the factors p_1, p_2, p_3, which is similar to that of the proof in [6] in the sense that the challenge is given in a subgroup of a composite order group and the factors are given to the simulator.
- $\mathcal{B}$ is given an instance of the subgroup decision assumption problem $INS_{SDA} = (N, G, G_T, e, g_1, g_3, X_1X_2, Y_2Y_3, T')$, where G is a bilinear group of order $N = p_1p_2p_3$ (p_1, p_2, p_3 are three distinct primes), G_{p_i} is the subgroup of order p_i in G (where $1 \le i \le 3$), $e : G \times G \to G_T$ is a bilinear map, $g_1, X_1 \in G_{p_1}$, $X_2, Y_2 \in G_{p_2}$, $g_3, Y_3 \in G_{p_3}$, and $\beta \in \{0,1\}$, $T' \in G_{p_1p_3}$ if $\beta = 1$, $T' \in G$ if $\beta = 0$.

The goal of $\mathcal{B}$ is to output a tuple $(c_r, w_r) \in \mathbb{Z}_{p_1} \times G_{p_1}$ satisfying $w_r = g_1^{\frac{1}{a+c_r}}$ for solving the q-SDH problem, and a bit $\beta' \in \{0,1\}$ to determine $T' \in G_{p_1p_3}$ or $T' \in G$. And $\mathcal{B}$ will make use of the $\mathcal{A}$ to break at least one of the q-SDH assumption and the subgroup decision assumption. $\mathcal{B}$ flips a random coin $\sigma \in \{0,1\}$ before playing the game with $\mathcal{A}$:

- if $\sigma = 0$, taking $INS_{SDH} = (N, G, G_T, e, p_1, p_2, p_3, g_1, g_1^a, \ldots, g_1^{a^q})$ as input, $\mathcal{B}$ set $A_i = g_1^{a^i}$ for $i = 0, 1, \ldots, q$.
- if $\sigma = 1$, taking $INS_{SDA} = (N, G, G_T, e, g_1, g_3, X_1 \cdot X_2, Y_2Y_3, T')$ as input, $\mathcal{B}$ chooses random $\tilde{a} \in \mathbb{Z}_N^*$ and computes $A_i = g_1^{\tilde{a}^i}$ for $i = 0, 1, \ldots, q$. In addition, unknown values $a = (\tilde{a}\ mod\ p_1) \in \mathbb{Z}_{p_1}^*$ is randomly chosen and $A_i = g_1^{\tilde{a}^i} = g_1^{a^i}$ for $i = 0, 1, \ldots, q$ are set implicitly.

$\mathcal{B}$ interacts with $\mathcal{A}$ as follows:

- **`Setup`** : $\mathcal{B}$ first chooses random $c_1, \ldots, c_{q'} \in \mathbb{Z}_N^*$. Let $f(y)$ be the polynomial $f(y) = \prod_{i=1}^{q'}(y + c_i)$.

Expand $f(y)$ and write $f(y)=\sum_{i=0}^{q'}\alpha_i y^i$ where $\{\alpha_i\}_{i\in\{0,1,\ldots,q'\}}\in\mathbb{Z}_N$ are the coefficients of the polynomial $f(y)$. $\mathcal{B}$ then computes $g\leftarrow\prod_{i=0}^{q'}(A_i)^{\alpha_i}=g_1^{f(a)}\in G_{p_1}$, $g^a\leftarrow\prod_{i=1}^{q'+1}(A_i)^{\alpha_{i-1}}=g_1^{f(a)\cdot a}$. $\mathcal{B}$ chooses $\alpha,b,\kappa\in\mathbb{Z}_N$, $R_0\in G_{p_3}$ randomly, and chooses a random value $u_i\in\mathbb{Z}_N$ for each attribute $i\in\mathcal{U}$. Let $INJ:G_N\rightarrow\mathbb{Z}_N$ be an efficiently computable injective encoding. $\mathcal{B}$ gives $pk=(N,g,g^a,g^b,g^\kappa, e(g,g)^\alpha,\{\mathcal{U}_i=g^{u_i}\}_{i\in\mathcal{U}})$ to $\mathcal{A}$.

- **Key Query**. $\mathcal{A}$ submits (id_i,S_i) to $\mathcal{B}$ to request a secret key. Assume it is the ith query. Noted $i\leq q'$, let $f_i(y)$ be the polynomial $f_i(y)=f(y)/(y+c_i)=\prod_{j=1,j\neq i}^{q'}(y+c_j)$. Expand $f_i(y)$ and write $f_i(y)=\sum_{j=0}^{q'-1}\eta_j y^j$. $\mathcal{B}$ then computes $\varrho_i\leftarrow\prod_{j=0}^{q'-1}(A_j)^{\eta_j}=g_1^{f_i(a)}=g_1^{\frac{f(a)}{a+c_i}}=g^{\frac{1}{a+c_i}}$. $\mathcal{B}$ randomly chooses $c,t,r,h'\in\mathbb{Z}_N$, $R,R',R'',R_0',R_0''\in G_{p_3}$, $R_i\in G_{p_3}$ for each $i\in S$, and computes $h=g^{h'}\in G_{p_1}$. The secret key $sk_{id,S}$ is set as follows: $\langle K=(\varrho_i)^\alpha g^{btr}g^{c\kappa}R=g^{\frac{\alpha}{a+c_i}}g^{btr}g^{c\kappa}R, K'=g^c h^r R', K''=g^{tr}R'', L=R_0^{id}h^{\kappa r}, L'=g^{ac}h^{ar}R_0', L''=g^{atr}R_0'', T=c_i,\{K_i=\mathcal{U}_i^{(a+c_i)tr}R_i\}_{i\in S},S\rangle$.
- **Key Forgery**. $\mathcal{A}$ outputs a decryption key sk_*. By ϵ_A we denote the event that $\mathcal{A}$ wins the game, i.e., sk_* is in the form of $sk_*=\langle K,K',K'',L,L',L'',T,\{K_i\}_{i\in S}\rangle$ and satisfies the key sanity check, and $T\notin\{c_1,c_2,\ldots,c_{q'}\}$.

If ϵ_A does not happen, $\mathcal{B}$ chooses a tuple $(c_r,w_r)\in\mathbb{Z}_{p_1}\times G_{p_1}$ and a bit $\beta'\in\{0,1\}$ randomly as its solution for the q-SDH problem and the subgroup decision assumption problem. If ϵ_A happens, using long division $\mathcal{B}$ writes the polynomial f as $f(y)=\gamma(y)(y+T)+\gamma_{-1}$ for some polynomial $\gamma(y)=\sum_{i=0}^{q'-1}\gamma_i y^i$ and some $\gamma_{-1}\in\mathbb{Z}_N$. $\mathcal{B}$ then computes $gcd(\gamma_{-1},N)$, the following two cases happen:

- Case 1: $gcd(\gamma_{-1},N)=1$.
  If $\sigma=0$, it implies that $\mathcal{B}$ made use of INS_{SDH} to interact with $\mathcal{A}$. $\mathcal{B}$ chooses a random bit $\beta'\in\{0,1\}$ as its output for the subgroup decision assumption problem. It computes a tuple $(c_r,w_r)\in\mathbb{Z}_{p_1}\times G_{p_1}$ as follows: Assuming $K'=g^c h^r K_2' K_3', K''=g^{tr}K_2''K_3''$ where $c,r,t\in\mathbb{Z}_N, K_2',K_2''\in G_{p_2}, K_3',K_3''\in G_{p_3}$ are unknown, we have $L'=g^{ac}h^{ar}L_2'L_3'$ (from (2) in the key sanity check algorithm), $L''=g^{atr}L_2''L_3''$ (from (3) in the key sanity check algorithm), $KL=g^{\frac{\alpha}{a+T}}g^{btr}g^{c\kappa}h^{\kappa r}K_2K_3$ (from (2), (3),(4) in the key sanity check algorithm), where $L_2',L_2'',K_2\in G_{p_2}, L_3',L_3'',K_3\in G_{p_3}$. $\mathcal{B}$ computes $1/\gamma_{-1}\ mod\ N$, and then computes $\theta\leftarrow((KL/((K')^\kappa(K'')^b))^{p_2p_3})^{(p_2p_3\alpha)^{-1}\ mod\ p_1}=g^{\frac{1}{a+T}}=g_1^{\gamma(a)}g_1^{\frac{\gamma_{-1}}{a+T}}$, $w_r\leftarrow(\theta\cdot\prod_{i=0}^{q'-1}A_i^{-\gamma_i})^{1/\gamma_{-1}}=g_1^{\frac{1}{a+T}}\in G_{p_1}$, $c_r\leftarrow T\ mod\ p_1\in\mathbb{Z}_{p_1}$. Note that since $e(g_1^a\cdot g_1^{c_r}, w_r)=e(g_1^a\cdot g_1^T, g_1^{\frac{1}{a+T}})=e(g_1,g_1)$, (c_r,w_r) is a solution for the q-SDH problem. If $\sigma=1$, it implies that $\mathcal{B}$ made use of INS_{SDA} to interact with $\mathcal{A}$. Since $gcd(\gamma_{-1},N)=1$ does not provide any useful information to $\mathcal{B}$, $\mathcal{B}$ chooses a tuple $(c_r,w_r)\in\mathbb{Z}_{p_1}\times G_{p_1}$ and a $\beta'\in\{0,1\}$ randomly as its solutions for the q-SDH problem and the subgroup decision assumption problem.
- Case 2: $gcd(\gamma_{-1},N)\neq 1$.
  If $\sigma=0$, it implies that $\mathcal{B}$ made use of INS_{SDH} to interact with $\mathcal{A}$. Since $gcd(\gamma_{-1},N)\neq 1$ does not provide any useful information to $\mathcal{B}$, $\mathcal{B}$ chooses a tuple $(c_r,w_r)\in\mathbb{Z}_{p_1}\times G_{p_1}$ and a bit $\beta'\in\{0,1\}$ randomly as its solutions for the q-SDH problem and the subgroup decision assumption problem.

  If $\sigma=1$, it implies that $\mathcal{B}$ made use of INS_{SDA} to interact with $\mathcal{A}$. $\mathcal{B}$ randomly chooses a tuple $(c_r,w_r)\in\mathbb{Z}_{p_1}\times G_{p_1}$ as its output for the q-SDH problem. $\mathcal{B}$ then determines the value of β' as follows:

  $\mathcal{B}$ gets two non-trivial factors $p,p'\in\mathbb{Z}_N$ such that $pp'=N$ using $gcd(\gamma_{-1},N)$. We have that $(p,p')\in\{(p_1,p_2p_3),(p_2p_3,p_1),(p_2,p_1p_3),(p_1\cdot p_3,p_2),(p_3,p_1p_2),(p_1p_2,p_3)\}$.
- If $g_1^p=1$ and $(Y_2Y_3)^{p'}=1$, it implies that $p=p_1$ and $p'=p_2p_3$. Otherwise, if $g_1^{p'}=1$ and $(Y_2Y_3)^p=1$, it implies that $p=p_2p_3$ and $p'=p_1$. Thus, $\mathcal{B}$ gets the value of p_1. It then computes $e((T')^{p_1},X_1X_2)$. If $e((T')^{p_1},X_1X_2)=1$, $\mathcal{B}$ sets $\beta'=1$, otherwise sets $\beta'=0$ (since $(T')^{p_1}\in G_{p_2p_3}$ if $T'\in G$ and $(T')^{p_1}\in G_{p_3}$ if $T'\in G_{p_1p_3}$).
- Otherwise, if $g_3^p=1$ and $(X_1X_2)^{p'}=1$, it implies that $p=p_3$ and $p'=p_1p_2$. Otherwise, if $g_3^{p'}=1$ and $(X_1X_2)^p=1$, it implies that $p=p_1p_2$ and $p'=p_3$. Thus, $\mathcal{B}$ gets the value of p_3. It then computes $e((T')^{p_3},Y_2Y_3)$. If $e((T')^{p_3},Y_2Y_3)=1$, $\mathcal{B}$ sets $\beta'=1$, otherwise sets $\beta'=0$ (since $(T')^{p_3}\in G_{p_1p_2}$ if $T'\in G$ and $(T')^{p_3}\in G_{p_1}$ if $T'\in G_{p_1p_3}$).
- Otherwise, if $g_3^p=1$, it implies that $p=p_1p_3$ and $p'=p_2$. Otherwise, if $g_3^{p'}=1$, it implies that $p=p_2$ and $p'=p_1p_3$. Thus, $\mathcal{B}$ gets the value of p_2. It then computes $(T')^{p_1p_3}$. If $(T')^{p_1p_3}=1$, $\mathcal{B}$ sets $\beta'=1$, otherwise sets $\beta'=0$ (since $(T')^{p_1p_3}\in G_{p_2}$ if $T'\in G$ and $(T')^{p_1p_3}=1$ if $T'\in G_{p_1p_3}$).

Now we evaluate $\mathcal{B}$'s advantage in breaking the q-SDH assumption and the subgroup decision assumption.

We denote $\varepsilon_{SDH}(c_r,w_r)$ be the event that (c_r,w_r) is a solution for the q-SDH problem. We note that when (c_r,w_r) is randomly chosen by $\mathcal{B}$, $\varepsilon_{SDH}(c_r,w_r)$ happens with negligible probability, for simplicity, say zero. And $e(g_1^a\cdot g_1^{c_r},w_r)=e(g_1,g_1)$ happens with probability 1 when (c_r,w_r) is chosen by $\mathcal{B}$ in the case of $(\mathcal{A}\ wins\wedge gcd(\gamma_{-1},N)=1\wedge\sigma=0)$.

$\mathcal{B}$ solves the q-SDH problem with probability

$$\begin{aligned}
&\Pr[\varepsilon_{SDH}(c_r, w_r)] \\
&= \Pr[\varepsilon_{SDH}(c_r, w_r)|\overline{\mathcal{A}\ wins}] \cdot \Pr[\overline{\mathcal{A}\ wins}] \\
&\quad + \Pr[\varepsilon_{SDH}(c_r, w_r)|(\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) = 1 \wedge \sigma = 0)] \\
&\quad \cdot \Pr[\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) = 1 \wedge \sigma = 0] \\
&\quad + \Pr[\varepsilon_{SDH}(c_r, w_r)|(\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) = 1 \wedge \sigma = 1)] \\
&\quad \cdot \Pr[\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) = 1 \wedge \sigma = 1] \\
&\quad + \Pr[\varepsilon_{SDH}(c_r, w_r)|(\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) \neq 1 \wedge \sigma = 0)] \\
&\quad \cdot \Pr[\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) \neq 1 \wedge \sigma = 0] \\
&\quad + \Pr[\varepsilon_{SDH}(c_r, w_r)|(\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) \neq 1 \wedge \sigma = 1)] \\
&\quad \cdot \Pr[\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) \neq 1 \wedge \sigma = 1] \\
&= \Pr[\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) = 1 \wedge \sigma = 0] + 0 + 0 + 0 + 0 \\
&= \Pr[\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) = 1] \cdot \Pr[\sigma = 0] \\
&= \frac{1}{2} \cdot \Pr[\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) = 1].
\end{aligned}$$

Note that $\beta' = \beta$ happens with probability $\frac{1}{2}$ when $\mathcal{B}$ chooses a bit $\beta' \in \{0, 1\}$. And $\beta' = \beta$ happens with probability 1 when $\mathcal{B}$ chooses a bit $\beta' \in \{0, 1\}$ in the case of $(\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) \neq 1 \wedge \sigma = 1)$

$$\begin{aligned}
&\Pr[\beta' = \beta] \\
&= \Pr[\beta' = \beta|\overline{\mathcal{A}\ wins}] \cdot \Pr[\overline{\mathcal{A}\ wins}] \\
&\quad + \Pr[\beta' = \beta|(\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) = 1 \wedge \sigma = 0)] \\
&\quad \cdot \Pr[\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) = 1 \wedge \sigma = 0] \\
&\quad + \Pr[\beta' = \beta|(\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) = 1 \wedge \sigma = 1)] \\
&\quad \cdot \Pr[\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) = 1 \wedge \sigma = 1] \\
&\quad + \Pr[\beta' = \beta|(\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) \neq 1 \wedge \sigma = 0)] \\
&\quad \cdot \Pr[\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) \neq 1 \wedge \sigma = 0] \\
&\quad + \Pr[\beta' = \beta|(\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) \neq 1 \wedge \sigma = 1)] \\
&\quad \cdot \Pr[\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) \neq 1 \wedge \sigma = 1] \\
&= \frac{1}{2}(1 - \epsilon) + \frac{1}{2}\Pr[\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) = 1 \wedge \sigma = 0] \\
&\quad + \frac{1}{2}\Pr[\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) = 1 \wedge \sigma = 1] \\
&\quad + \frac{1}{2}\Pr[\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) \neq 1 \wedge \sigma = 0] \\
&\quad + \Pr[\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) \neq 1 \wedge \sigma = 1] \\
&= \frac{1}{2}(1 - \epsilon) + \frac{1}{2}\Pr[\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) \neq 1 \wedge \sigma = 1] \\
&\quad + \frac{1}{2}\Pr[\mathcal{A}\ wins] \\
&= \frac{1}{2} + \frac{1}{4}\Pr[\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) \neq 1].
\end{aligned}$$

Thus, $\mathcal{B}$'s advantage in breaking the q-SDH assumption is $Adv_{SDH} = \Pr[\varepsilon_{SDH}(c_r, w_r)] = \frac{1}{2} \cdot \Pr[\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) = 1]$. $\mathcal{B}$'s advantage in breaking the subgroup decision assumption is $Adv_{SDA} = |\Pr[\beta' = \beta] - \frac{1}{2}| = \frac{1}{4}\Pr[\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) \neq 1]$. Observe that $Adv_{SDH} + Adv_{SDA} = \frac{\epsilon}{4} + \frac{1}{4}\Pr[\mathcal{A}\ wins \wedge gcd(\gamma_{-1}, N) = 1] \geq \frac{\epsilon}{4}$. We have that at least one of Adv_{SDH}, Adv_{SDA} is $\geq \frac{\epsilon}{8}$ according to Dirichlets drawer principle. □

5.4 Key Sanity Check Proof

We adopt the proof method from [1], [33] to give the key sanity check proof of our T-CPABE-pbCom system.

Theorem 3. *The advantage of an adversary in the key sanity check game is negligible for our T-CPABE-pbCom system.*

Proof. Let the output of an attacker $\mathcal{A}$ be the public parameters pk, a ciphertext $\mathcal{T}$, two different secret keys $sk_{id,S} = \langle K, K', K'', L, L', L'', T, \{K_i\}_{i\in S}\rangle$ and $\tilde{sk}_{id,S} = \langle \tilde{K}, \tilde{K}', \tilde{K}'', \tilde{L}, \tilde{L}', \tilde{L}'', \tilde{T}, \{\tilde{K}_i\}_{i\in S}\rangle$. $\mathcal{A}$ wins implies that the following conditions (as defined in the key sanity check game) are all fulfilled.

Conditions $(1) - (5)$:

(1) $\texttt{KeySanityCheck}(pk, sk_{id,S}) \rightarrow 1$.
(2) $\texttt{KeySanityCheck}(pk, \tilde{sk}_{id,S}) \rightarrow 1$.
(3) $\texttt{Decrypt}(pk, sk_{id,S}, \mathcal{T}) \neq \perp$.
(4) $\texttt{Decrypt}(pk, \tilde{sk}_{id,S}, \mathcal{T}) \neq \perp$.
(5) $\texttt{Decrypt}(pk, sk_{id,S}, \mathcal{T}) \neq \texttt{Decrypt}(pk, \tilde{sk}_{id,S}, \mathcal{T})$.

Condition (1) implies

(1) $K, K', K'', L, L', L'', \{K_i\}_{i\in S} \in G, T \in Z_N$.
(2) $e(L', g) = e(K', g^a) \neq 1$.
(3) $e(L'', g) = e(K'', g^a) \neq 1$.
(4) $e(KL, g^a \cdot g^T) = e(g, g)^{\alpha} e((K')^T L', g^{\kappa}) \cdot e((K'')^T L'', g^b) \neq 1$.
(5) $\exists i \in S,\ s.t.\ e(K_i, g) = e(\mathcal{U}_i, (K'')^T L'') \neq 1$.
(6) $T = INJ(L)$.

Similarly, condition (2) implies

(1) $\tilde{K}, \tilde{K}', \tilde{K}'', \tilde{L}, \tilde{L}', \tilde{L}'', \{\tilde{K}_i\}_{i\in S} \in G, \tilde{T} \in Z_N$.
(2) $e(\tilde{L}', g) = e(\tilde{K}', g^a) \neq 1$.
(3) $e(\tilde{L}'', g) = e(\tilde{K}'', g^a) \neq 1$.
(4) $e(\tilde{K}\tilde{L}, g^a \cdot g^{\tilde{T}}) = e(g, g)^{\alpha} e((\tilde{K}')^{\tilde{T}} \tilde{L}', g^{\kappa}) \cdot e((\tilde{K}'')^{\tilde{T}} \tilde{L}'', g^b) \neq 1$.
(5) $\exists i \in S,\ s.t.\ e(\tilde{K}_i, g) = e(\tilde{\mathcal{U}}_i, (\tilde{K}'')^{\tilde{T}} \tilde{L}'') \neq 1$.
(6) $\tilde{T} = INJ(\tilde{L})$.

From conditions (1) and (3), we have

$$\begin{aligned}
E &= \Pi_{\rho(j)\in S}(e(C_{j,1}, (K'')^T L'') e(C_{j,2}, K_{\rho(j)}))^{\omega_j}, F \\
&= e((C_0)^T C_1, KL) e(C_2, (K')^T L')^{-1}, F/E = e(g, g)^{\alpha s}, m \\
&= C/(F/E).
\end{aligned}$$

Similarly, from conditions (2) and (4), we have

$$\begin{aligned}
\tilde{E} &= \Pi_{\rho(j)\in S}(e(C_{j,1}, (\tilde{K}'')^{\tilde{T}} \tilde{L}'') e(C_{j,2}, \tilde{K}_{\rho(j)}))^{\omega_j}, \tilde{F} \\
&= e((C_0)^{\tilde{T}} C_1, \tilde{K}\tilde{L}) e(C_2, (\tilde{K}')^{\tilde{T}} \tilde{L}')^{-1}, \tilde{F}/\tilde{E} = e(g, g)^{\alpha s}, m \\
&= C/(\tilde{F}/\tilde{E}).
\end{aligned}$$

From conditions $(1) - (4)$, we have

$$F/E = e(g, g)^{\alpha s} = \tilde{F}/\tilde{E}, m = C/(F/E) = C/(\tilde{F}/\tilde{E}) \quad (*).$$

However, condition (5) implies that $C/(F/E) \neq C/(\tilde{F}/\tilde{E})$, which contradicts to $(*)$. Thus $\mathcal{A}$ wins the game only with negligible probability. □

6 Performance Evaluations

6.1 Theoretical Analysis

Table 1 gives the comparison between our work and the conventional CP-ABE in [23], [24] in terms of performance.

TABLE 1
Performance Comparison with Conventional CP-ABE[1]

	PubKS	PriKS	CS	PCD
[23]	$\|u\|+3$	$\|S\|+2$	$2l+2$	$2\|I\|+1$
[24]	$\|u\|+4$	$\|S\|+3$	$2l+3$	$2\|I\|+2$
this work	$\|u\|+5$	$\|S\|+7$	$2l+4$	$2\|I\|+2$

[1]*PubKS stands for public key size, PriKS stands for private key size, CS stands for ciphertext size, PCD stands for pairing computation in decryption. Let $|u|$ be the size of the attribute universe, $|S|$ the size of the attribute set of a private key, l the size of an access policy, and $|I|$ the number of attributes in a decryption key that satisfies a ciphertext's access policy.*

In particular, our system achieves the functionality of white-box traceability from commitment (with zero additional storage overhead for tracing) with the price of adding only one element in public key and ciphertext, four elements in private key, and one pairing computation in decryption. Clearly, the complexity of the proposed system is acceptable in practice.

Tables 2 and 3 give the comparisons between our work and some other related work in terms of features (i.e., Traceability, etc.) and performance. Compared with other related work, our proposed system only sacrifices tiny size of the private key and the ciphertext to achieve white-box traceability. In particular, the new system needs no storage for tracing, and still keeps other important features (i.e., supporting any monotone access structures, fully secure, standard model). This makes the new system more practical for applications (such as cloud storage service).

6.2 Experimental Analysis

In this section, we evaluate the performance of the proposed systems in Section 5. The experiments are performed on a laptop using the Intel Core i5-5200U at a frequency of 2.20 GHz with 4 GB memory and Windows 7 operation system with Service Pack 1. We use the pairing-based cryptography library [29] with type A1 curve to realize the proposed systems. The programming language is Java with JDK32-1.6.0 and JPBC-2.0.0 [10].

In CP-ABE systems, the complexity of ciphertext policy impacts both the encryption time and the decryption time. To illustrate this, we generate ciphertext policies in the form of (S_1 and S_2...and S_l) to simulate the worst situation, where S_i is an attribute. We aim to evaluate the efficiency of our T-CPABE-pbCom system by comparing the total time taken during each stage with the original CP-ABE in [24] which does not consider the traceability issue. As depicted in Fig. 1, we examine the time cost of executing individual stage (including the Setup Time, the KeyGen Time, Encrypt Time and the Decrypt Time). It is not surprising to see that our system takes more time since we consider the traceability issue. Figs. 1a, 1b, 1c, and 1d illustrate that our T-CPABE-pbCom system achieves the white-box traceability without introducing significant overhead compared to the original CP-ABE in [24]. We note that the method to add the traceability issue we introduced is general (which is also applicable to other CP-ABE systems), it is easy to apply our technique to obtain a more efficient system.

TABLE 2
Features Comparison with Other Related Work[1]

	T	TP	MAS	FS	SM
[26]	√	none	×	×	×
[25]	√	none	×	×	√
[28]	√	linear	√	√	√
[27]	√	sub-linear	√	√	√
[31]	√	constant	√	×	√
[33]	√	constant	√	×	√
this work	√	none	√	√	√

[1]*T stands for traceability, TP stands for tracing price, MAS stands for supporting any monotone access structures, FS stands for fully secure, and SM stands for standard model.*

TABLE 3
Performance Comparison with Other Related Work[1,2]

	PubKS	PriKS	CS	PCD	SCT[3]
[26]	$\|u\|+4$	$\|S\|+3$	$l+3$	2	0
[25]	$3\|u\|+3\rho+5$	$4\|S\|+3\rho$	$2l+4\rho+2$	$3\|I\|+3\rho$	0
[28]	$\|u\|+4$	$\|S\|+4$	$2l+3$	$2\|I\|+1$	$\mathcal{N}$
[27]	$\|u\|+3+4\sqrt{\mathcal{N}}$	$\|S\|+4$	$2l+17\sqrt{\mathcal{N}}$	$2\|I\|+10$	$\sqrt{\mathcal{N}}$
[31]	7	$2\|S\|+4$	$3l+3$	$3\|I\|+1$	$INS_{(t,n)}$
[33]	7	$2\|S\|+4$	$3l+3$	$3\|I\|+1$	$INS_{(t,n)}$
this work	$\|u\|+5$	$\|S\|+7$	$2l+4$	$2\|I\|+2$	0

[1]*PubKS stands for public key size, PriKS stands for private key size, CS stands for ciphertext size, PCD stands for pairing computation in decryption, SCT stands for storage cost for tracing. Let $|u|$ be the size of the attribute universe, $|S|$ the size of the attribute set of a private key, l the size of an access policy, $|I|$ the number of attributes in a decryption key that satisfies a ciphertext's access policy, and ρ the bit length of the global identity in [25].*

[2]*In [28], the storage cost for tracing grows linearly in $\mathcal{N}$, where $\mathcal{N}$ is the number of users in the system. In [27], the storage cost for tracing grows sub-linearly in $\mathcal{N}$. And in [31], [33], the storage cost for tracing is an instance of Shamir's (t,n) threshold scheme $INS_{(t,n)}$ including (at least) $t-1$ points on a polynomial $f(x)$ and $f(0)$.*

7 Extensions

7.1 Transform from Composite-Order Setting to Prime-Order Setting

The construction of our proposed system is built on composite order group. In composite order groups, security typically relies on the hardness of factoring the group order. And this requires the use of large group orders, which results in considerably slower pairing operations. Though composite order bilinear groups have appealing features, it is desirable to obtain the same functionalities in prime-order groups [22].

We can extend our T-CPABE-pbCom system to a prime-order setting system using the technique used in [8], [22]. Specifically, we can utilize dual pairing vector space framework initiated by Okamoto and Takashima [34], [35] to formulate an assumption in prime order groups that can be used to mimic the effect of the general subgroup decision assumption in composite order groups [22]. And as [22], we can obtain a toolkit for turning our composite order construction into prime order constructions that can be proven secure from DLIN. As showed in the full version of [24], the composite order construction (which is the underling construction of our system) can be transformed to a prime order analog using the translation techniques (based on DPVS) developed in [22]. Note that the translation techniques which based on DPVS developed in [22] are typically

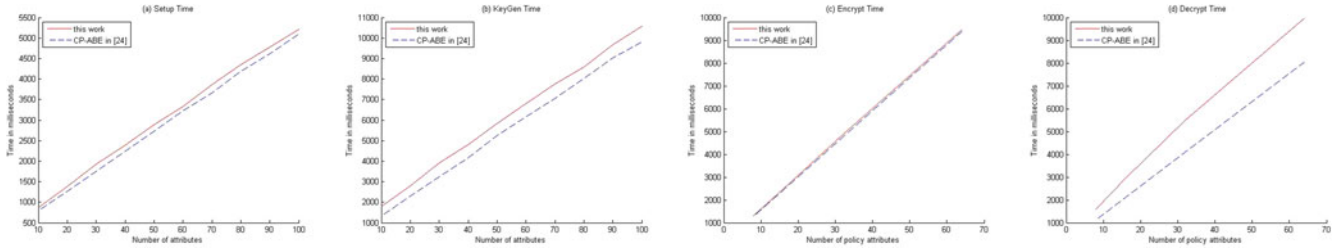

Fig. 1. Experimental results.

applicable for composite order constructions relying on the canceling property and proven secure from variants of the subgroup decision assumption. It is worth noting that both the underlying CP-ABE construction of our system and the extractable commitments we employed are basically and crucially relying on the canceling property and proven secure from variants of the subgroup decision assumption, which are the constructions supported by the translation techniques (based on DPVS) in [22]. Furthermore, we can adopt the technique introduced in [8] for simulating composite-order groups in prime-order ones to improve the efficiency of the resulting system. Concretely, using the novel idea and new encoding technique introduced in [8], we do not have to simulate all of the structure in composite-order groups, thus leading to better concrete efficiency. We note that the method introduced above is general and we leave the detailed transformation (where some detailed problems may arise) as a promising future direction.

7.2 Revocable T-CPABE-pbCom System

The T-CPABE-pbCom system provides an effective way to track the malicious cloud users leaking their access credentials for profits. And for those users who are identified to be malicious need to be revoked from the system. This evokes another important issue to deal with: how to revoke the identified malicious users.

We can extend our T-CPABE-pbCom system to a revocable system for cloud storage service by using the technique introduced in [37]. Specifically, we can utilize the technology of ciphertext delegation and piecewise key generation used in [37] to achieve the functionality of revoke. We note that since our proposed system makes use of commitment in the `Trace` algorithm, the system need not maintain an identity table T which contains all users identifies. This brings an advantage that the system need not maintain (i.e., update) the table T when some malicious users are revoked.

8 Conclusion and Future Work

In this work, we first presented two different kinds of non-interactive pairing-friendly commitments for traitor tracing. Then we proposed a white-box traceable CP-ABE system for cloud storage service from one of the presented commitments. The proposed system is proved fully white-box traceability and fully secure in the standard model. As far as we know, it is the first white-box traceable CP-ABE system for cloud storage service from non-interactive traceable pairing-friendly commitment with perfectly binding key, which supports any monotone access structures and achieves fully secure in the standard model.

Note that in our security model of white-box traceability, the secret key forged by the adversary needs to pass the key sanity check. In our concrete construction, a secret key needs to be in a specific form to pass the key sanity check. To what extent to define a forged secret key (output by the adversary) that can always be able to decrypt (the corresponding) ciphertexts might be challenging, we leave it as interesting future work to obtain the ideal definition of key sanity check. Also note that there exists a relatively stronger notion called black-box traceability: the leakage of the user is the decryption equipment instead of its decryption key [27], [31]. And the malicious users could hide the decryption algorithm and the decryption keys by tweaking it. Therefore, the decryption keys and decryption algorithms are both not well-formed, and the key sanity check fails in that case. We leave it as interesting future work to obtain an efficient fully secure black-box traceable CP-ABE from pairing-friendly commitments.

Acknowledgments

The authors are grateful to the anonymous referees for their invaluable suggestions. We would like to thank Dr. Kaitai Liang for useful discussions, and Mr. Cong Zuo for many helps on the experiments. This work was supported by the Natural Science Foundation of China (Grant No. 61632012, 61672239, 61371083, 61373154, 61411146001 and 61402282), the Prioritized Development Projects of the Specialized Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20130073130004) and the Shanghai Science and Technology Commission Project (Grant No. 16511101400, 14YF1410400). Zhenfu Cao and Xiaolei Dong are the corresponding authors.

References

[1] M. H. Au, Q. Huang, J. K. Liu, W. Susilo, D. S. Wong, and G. Yang, "Traceable and retrievable identity-based encryption," in *Proc. 6th Int. Conf. Appl. Cryptography Netw. Secur.*, 2008, pp. 94–110.

[2] A. Beimel, "Secure schemes for secret sharing and key distribution," PhD dissertation, Israel Inst. Technol., Technion, Haifa, Israel, 1996.

[3] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy*, 2007, pp. 321–334.

[4] D. Boneh and X. Boyen, "Short signatures without random oracles," in *Proc. Int. Conf. Theory Appl. Cryptographic Techn. Advances Cryptology*, 2004, pp. 56–73.

[5] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Proc. 2nd Int. Conf. Theory Cryptography*, 2005, pp. 325–341.

[6] D. Boneh, A. Sahai, and B. Waters, "Fully collusion resistant traitor tracing with short ciphertexts and private keys," in *Advances in Cryptology*. Berlin, Germany: Springer, 2006, pp. 573–592.

[7] X. Boyen, Q. Mei, and B. Waters, "Direct chosen ciphertext security from identity-based techniques," in *Proc. 12th ACM Conf. Comput. Commun. Secur.*, 2005, pp. 320–329.

[8] J. Chen, R. Gay, and H. Wee, "Improved dual system abe in prime-order groups via predicate encodings," in *Advances Cryptology*. Berlin, Germany: Springer, 2015, pp. 595–624.
[9] R. Cramer and V. Shoup, "Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack," *SIAM J. Comput.*, vol. 33, no. 1, pp. 167–226, 2003.
[10] A. De Caro and V. Iovino, "JPBC: Java pairing based cryptography," in *Proc. 16th IEEE Symp. Comput. Commun.*, 2011, pp. 850–855.
[11] R. R. Farashahi, B. Schoenmakers, and A. Sidorenko, "Efficient pseudorandom generators based on the DDH assumption," in *Public Key Cryptography*. Berlin, Germany: Springer, 2007, pp. 426–441.
[12] O. Goldreich, *Foundation of Cryptography*. Cambridge, U.K.: Cambridge Univ. Press, 2001.
[13] S. Goldwasser and S. Micali, "Probabilistic encryption," *J. Comput. Syst. Sci.*, vol. 28, no. 2, pp. 270–299, 1984.
[14] V. Goyal, S. Lu, A. Sahai, and B. Waters, "Black-box accountable authority identity-based encryption," in *Proc. 15th ACM Conf. Comput. Commun. Secur.*, 2008, pp. 427–436.
[15] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, 2006, pp. 89–98.
[16] J. Groth, R. Ostrovsky, and A. Sahai, "New techniques for noninteractive zero-knowledge," *J. ACM*, vol. 59, no. 3, 2012, Art. no. 11.
[17] M. J. Hinek, S. Jiang, R. Safavi-Naini, and S. F. Shahandashti, "Attribute-based encryption with key cloning protection," *IACR Cryptology*, vol. 2008, 2008, Art. no. 478.
[18] J. Katz and D. Schröder, "Tracing insider attacks in the context of predicate encryption schemes," *ACITA*, 2011. [Online] Available at: https://www.usukita.org/node/1779.
[19] A. Kiayias and Q. Tang, "How to keep a secret: Leakage deterring public-key cryptosystems," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 943–954.
[20] A. Kiayias and Q. Tang, "Traitor deterring schemes: Using bitcoin as collateral for digital content," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 231–242.
[21] E. Kiltz, "Chosen-ciphertext secure key-encapsulation based on gap hashed Diffie-Hellman," in *Public Key Cryptography*. Berlin, Germany: Springer, 2007, pp. 282–297.
[22] A. Lewko, "Tools for simulating features of composite order bilinear groups in the prime order setting," in *Advances in Cryptology*. Berlin, Germany: Springer, 2012, pp. 318–335.
[23] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Advances in Cryptology*. Berlin, Germany: Springer, 2010, pp. 62–91.
[24] A. Lewko and B. Waters, "New proof methods for attribute-based encryption: Achieving full security through selective techniques," in *Advances in Cryptology*. Berlin, Germany: Springer, 2012, pp. 180–198.
[25] J. Li, Q. Huang, X. Chen, S. S. M. Chow, D. S. Wong, and D. Xie, "Multi-authority ciphertext-policy attribute-based encryption with accountability," in *Proc. 6th ACM Symp. Inf. Comput. Commun. Secur.*, 2011, pp. 386–390.
[26] J. Li, K. Ren, and K. Kim, "A2BE: Accountable attribute-based encryption for abuse free access control," *IACR Cryptology*, vol. 2009, 2009, Art. no. 118.
[27] Z. Liu, Z. Cao, and D. S. Wong, "Blackbox traceable CP-ABE: How to catch people leaking their keys by selling decryption devices on ebay," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 475–486.
[28] Z. Liu, Z. Cao, and D. S. Wong, "White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures," *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 1, pp. 76–88, Jan. 2013.
[29] B. Lynn, "The pairing-based cryptography library," http://crypto.stanford.edu/pbc/.
[30] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, 2010.
[31] J. Ning, Z. Cao, X. Dong, L. Wei, and X. Lin, "Large universe ciphertext-policy attribute-based encryption with white-box traceability," in *Computer Security*. Berlin, Germany: Springer, 2014, pp. 55–72.
[32] J. Ning, X. Dong, Z. Cao, and L. Wei, "Accountable authority ciphertext-policy attribute-based encryption with white-box traceability and public auditing in the cloud," in *Computer Security*. Berlin, Germany: Springer, 2015, pp. 270–289.
[33] J. Ning, X. Dong, Z. Cao, L. Wei, and X. Lin, "White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 6, pp. 1274–1288, Jun. 2015.
[34] T. Okamoto and K. Takashima, "Homomorphic encryption and signatures from vector decomposition," in *Pairing-Based Cryptography*. Berlin, Germany: Springer, 2008, pp. 57–74.
[35] T. Okamoto and K. Takashima, "Hierarchical predicate encryption for inner-products," in *Advances in Cryptology*. Berlin, Germany: Springer, 2009, pp. 214–231.
[36] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Advances in Cryptology*. Berlin, Germany: Springer, 1992, pp. 129–140.
[37] A. Sahai, H. Seyalioglu, and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in *Advances in Cryptology*. Berlin, Germany: Springer, 2012, pp. 199–217.
[38] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology*. Berlin, Germany: Springer, 2005, pp. 457–473.
[39] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.
[40] J. Staddon, P. Golle, M. Gagné, and P. Rasmussen, "A content-driven access control system," in *Proc. 7th Symp. Identity Trust Internet*, 2008, pp. 26–35.

Jianting Ning received the BS and MS degrees from Yangzhou University, in 2010 and 2013, respectively. He is currently working toward the PhD degree in the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research interests include cryptography and information security, in particular, public key encryption, attribute-based encryption, non-interactive proof systems for bilinear groups, and zero-knowledge proof.

Zhenfu Cao (SM'10) received the BSc degree in computer science and technology and the PhD degree in mathematics from Harbin Institute of Technology, Harbin, China, in 1983 and 1999, respectively. His research interests mainly include number theory, cryptography and information security. Up to now (since 1981), more than 400 academic papers have been published in Journals or conferences. He is currently a distinguished professor with East China Normal University, China. He also serves as a member of the expert panel of the National Nature Science Fund of China. He is actively involved in the academic community, serving as committee/co-chair and program committee member of several international conference committees. He is an associate editor of the *Computers and Security* (Elsevier) and the *Security and Communication Networks* (John Wiley), an editorial board member of the *Fundamenta Informaticae* (IOS) and the *Peer-to-Peer Networking and Applications* (Springer-Verlag), and guest editor of the *Wireless Communications and Mobile Computing* (Wiley), and the *IEEE Transactions on Parallel and Distributed Systems*, etc. He has received a number of awards, including the Youth Research Fund Award of the Chinese Academy of Science in 1986, the Ying-Tung Fok Young Teacher Award in 1989, the National Outstanding Youth Fund of China in 2002, the Special Allowance by the State Council in 2005 etc. He is also the leaders of Asia 3 Foresight Program (61161140320) and the key project (61033014) of National Natural Science Foundation of China. He is a senior member of the IEEE.

Xiaolei Dong received the doctorate degree from Harbin Institute of Technology, she is working toward the post-doctoral degree at Shanghai Jiao Tong University, from September 2001 to July 2003. She is a distinguished professor with East China Normal University. Her primary research interests include number theory, cryptography, and trusted computing, etc. Her "Number Theory and Modern Cryptographic Algorithms" project won the first prize of China University Science and Technology Award in 2002. Her "New Theory of Cryptography and Some Basic Problems" project won the second prize of Shanghai Nature Science Award in 2007. Her "Formal Security Theory of Complex Cryptographic System and Applications" won the second prize of Ministry of Education Natural Science Progress Award in 2008. Currently, she hosts a number of research projects supported by the National Basic Research Program of China (973 program), the special funds on information security of the National Development and Reform Commission and National Natural Science Foundation of China, etc.

Lifei Wei received the BSc and MSc degrees in applied mathematics from the University of Science and Technology Beijing, Beijing, China, in 2005 and 2007, respectively, and the PhD degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2013. He is currently an assistant professor in the Department of Information and Computing Sciences, Shanghai Ocean University. His research interests include applied cryptography, cloud computing, and wireless network security.