White-Box Traceable Ciphertext-Policy Attribute-Based Encryption Supporting Any Monotone Access Structures

Zhen Liu, Zhenfu Cao, *Senior Member, IEEE*, and Duncan S. Wong

***Abstract*—In a ciphertext-policy attribute-based encryption (CP-ABE) system, decryption keys are defined over attributes shared by multiple users. Given a decryption key, it may not be always possible to trace to the original key owner. As a decryption privilege could be possessed by multiple users who own the same set of attributes, malicious users might be tempted to leak their decryption privileges to some third parties, for financial gain as an example, without the risk of being caught. This problem severely limits the applications of CP-ABE. Several traceable CP-ABE (T-CP-ABE) systems have been proposed to address this problem, but the expressiveness of policies in those systems is limited where only AND gate with wildcard is currently supported. In this paper we propose a new T-CP-ABE system that supports policies expressed in any monotone access structures. Also, the proposed system is as efficient and secure as one of the best (non-traceable) CP-ABE systems currently available, that is, this work *adds* traceability to an existing expressive, efficient, and secure CP-ABE scheme *without* weakening its security or setting any particular trade-off on its performance.**

***Index Terms*—Attribute-based encryption, ciphertext-policy, traceability.**

I. Introduction

THE notion of Attribute-Based Encryption (ABE) was introduced by Sahai and Waters [1], as a generalization of fuzzy Identity-Based Encryption (IBE) [2], [3]. In [4] Goyal *et al.* proposed an expressive Key-Policy ABE (KP-ABE) scheme, and formalized the notion of Ciphertext-Policy ABE (CP-ABE). In a CP-ABE system, each user is issued a decryption key by an authority according to the attributes he possesses, and the encryptor decides what attributes the eligible receivers should have by encrypting the messages with an access policy defined over some attributes. If and only if a user's attributes satisfy the access policy of a ciphertext, he can decrypt the ciphertext. KP-ABE is reversed in that each ciphertext is described by a set of attributes and each user is issued a decryption key according to an access policy by an authority. Not only does ABE (especially CP-ABE) provide a new promising tool for implementing fine-grained access control over encrypted data, but also has it attracted much attention in the research community, and a series of work [4]–[11] has been done to achieve better expressiveness, efficiency or security. In particular, as an example of elegant work, the CP-ABE system due to Lewko *et al.* [10] is expressive (realizing any monotone access structures), efficient and provably (and fully) secure.

However, there is a major issue awaiting to be solved that limits the applications of ABE to date. In an ABE (CP-ABE as an example) system, the decryption keys are not associated with individuals, i.e., the decryption keys are defined over attributes shared by multiple users and are not uniquely linked to users' identification information (e.g., identities). This is the foundation of ABE to implement the efficient one-to-many encryption and the expressive access control (i.e., the encryptor does not need to know exactly or explicitly specify who the receivers are, and can decide the target receivers by encrypting the data with an access policy over descriptive attributes), but this also introduces the problem that the decryption keys are non-traceable: given a decryption key, it may be impossible to find out the original key owner, because there are always many users sharing the corresponding attributes. Imagine that a company deploys a CP-ABE system, and two employees, Bob and Tom, have their decryption keys corresponding to attribute sets "{Bob, Senior Engineer, Department of Research}" and "{Tom, Senior Engineer, Department of Research}" respectively. For efficient one-to-many encryption and expressive access control, data would generally be generated under an access policy with role-based descriptive attributes such as "((Senior Engineer **AND** Department of Research) **OR** Manager)" rather than with the identity-related attributes such as "(Bob **OR** Tom)". In an expressive CP-ABE system such as [9], [10], both Bob and Tom can delegate a decryption key for attribute set "{Senior Engineer, Department of Research}". Then when a decryption key for such an attribute set is up for sale, who did it? Bob or Tom?

In traditional Public Key Encryption (PKE) and IBE, the decryption privilege of a decryption key is exclusive to the key

Manuscript received May 14, 2012; revised July 28, 2012; accepted September 18, 2012. Date of publication October 09, 2012; date of current version December 26, 2012. The work of Z. Liu and Z. Cao was supported in part by the Key Program of the National Natural Science Foundation of China under Grant 61033014, in part by the National Natural Science Foundation of China under Grant 60970110 and Grant 60972034, in part by the National Natural Science Foundation of China A3 Foresight Program under Grant 61161140320, and in part by the National Fundamental Research Development Program of China (973) under Grant 2012CB723401. The work of D. S. Wong was supported by a grant from the RGC of the HKSAR, China (Project CityU 123511). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Michel Abdalla.

Z. Liu is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong (e-mail: liuzhen@sjtu.edu.cn; zhenliu7@student.cityu.edu.hk).

Z. Cao is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: zfcao@cs.sjtu.edu.cn).

D. S. Wong is with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong (e-mail: duncan@cityu.edu.hk).

Digital Object Identifier 10.1109/TIFS.2012.2223683

1556-6013/$31.00 © 2012 IEEE

owner and only the key owner's privacy is compromised by the leakage of the decryption key, so that the users are reluctant in leaking their decryption keys. However, in ABE, the decryption privilege of a decryption key is shared by multiple users who possess the corresponding attributes, so that any malicious owner of a decryption key would have the intention or be very willing to leak partial or even his entire decryption privilege for financial interest or any other incentive, especially when there is no risk of getting caught. We refer to this issue as *Malicious Key Delegation*. As a result, this is very crucial for a practical ABE system to support the traceability of decryption keys. Of course, this problem can be addressed by using tamper-resistant hardware to store the decryption keys, but it will result in very restricted applications. In this paper, we focus on designing an expressive CP-ABE scheme that inherently resists *malicious key delegation*.

The problem of building a secure ABE scheme resisting malicious key delegation has recently been studied in [12]–[15]. However, as we will review shortly, there is no traceable CP-ABE system which is as expressive (i.e., supporting any monotone access structures), efficient, and secure as the (non-traceable) CP-ABE system in [10]. [14] is a KP-ABE system. [12], [13], [15] are CP-ABE systems, but the ciphertext-policy can only be **AND** gate with wildcard, and the size of decryption key or/and ciphertext is linear in the size of the entire attribute universe. In addition, [12] relies on a trusted third party that interacts with a user each time when the user wants to decrypt a ciphertext, and which makes the system less practical.

A. Our Results

In this paper, we focus on building a white-box traceable CP-ABE as of [12] and [13] do and target to make contributions on adaptive security as well as full expressiveness, i.e., supporting any monotone access structures, as we notice that both [12] and [13] do not support adaptive security or full expressiveness. The term *White-Box Traceability* means that the cheating user is selling his/her decryption key or masqueraded/modified decryption key to some buyers, and the buyers will use the existing decryption equipment or module to perform decryption with the purchased key. This notion is similar to the original traceability proposed by Goyal [16] and referred to as white-box traceability by Goyal *et al.* in [17] in the setting of identity-based encryption. Note that a stronger notion is called black-box traceability[1]: the cheating user sells to the buyers a decryption equipment (which is built from his decryption key and tweaked decryption algorithm). However, there is no CP-ABE available that supports traceability (even in the white-box model) and high expressiveness simultaneously. Hence in this paper we focus on achieving white-box traceability with high expressiveness, that is, supporting any monotone access structures, and with adaptive security. We believe that this work will contribute as the first step towards building a high expressive, adaptively secure black-box traceable CP-ABE system, which is considered as one of our future work.

TABLE I
COMPARISON WITH EXISTING WORK

	[12]	[13]	[15]	this paper
Adaptively Secure	×	×	×	√
Supporting Any Monotone Access Structures[1]	×	×	×	√
Efficient[2]	×	×	×	√
Reduce Trust on Authority[3]	NA	√	NA	√
Black Box Tracable[4]	×	×	√	×
No Flaws Identified[4]	√	√	×	√

[1] In [12], [13], [15], the ciphertext policy can only be a single **AND** gate with wildcard.
[2] In [12], [13], [15], the ciphertext size is linear in the size of the entire attribute universe. [12] relies on a trusted third party that interacts with a user each time when the user wants to decrypt a ciphertext, and which makes the system less practical.
[3] [12], [15] did not consider the problem of reducing trust on the authority.
[4] Although [15] made attempts to achieve black-box traceability, there are flaws (shown in Appendix A) that make the traceability not be able to achieve as claimed.

In this paper, we propose a new CP-ABE system that supports traceability of malicious users who leaked their decryption privileges. This *Traceable* CP-ABE does not weaken the expressiveness or efficiency when compared with the most efficient conventional (non-traceable) CP-ABE systems currently available for high expressiveness (i.e., supporting any monotone access structures). In particular, each decryption key in our system can be traced to its owner, the ciphertext policy can be any monotone access structures, the decryption key size grows linearly with the number of corresponding attributes, and the ciphertext size grows linearly with the size of corresponding access structure. To the best of our knowledge, this is the first CP-ABE scheme that supports both traceability (of malicious users) and high expressiveness (i.e., supporting any monotone access structures) simultaneously. Also, the system can be shown to be adaptively secure in the standard model.

We also propose an extended version for our Traceable CP-ABE system. In this extension, there are two authorities, one for generating tracing information, and the other one for issuing decryption keys to users, while no single authority is able to independently generate decryption keys. This extended system can therefore, reduce the trust on each individual authority, and this technique can be viewed to be orthogonal to that of using threshold policy between multiple authorities as proposed in [3].

In Table I, we compare this work with some existing results in the literature that are related to Traceable CP-ABE.

B. Our Techniques

We now give a brief review on some existing ABE systems and explain why it is not obvious to extend any of them to a variant for supporting traceability. We notice that any straightforward extensions of the existing ABE systems would be susceptible to attacks which allow a malicious user to avoid being traced. In this section, we also introduce our ideas of constructing a secure and efficient Traceable CP-ABE before giving the full details in Section IV.

Besides achieving traceability, our goal is to maintain the expressiveness, efficiency and also the security of our newly proposed Traceable CP-ABE system when it is compared with the

[1] The white-box model may also be referred to as the "non-black-box" model.

two most recent CP-ABE systems [9], [10] that are expressive, efficient and provably secure. In [9], [10], the decryption key is in the form of $\mathsf{SK} = (K = g^{\alpha}h^{t}, L = g^{t}, K_x = U_x^{t}\ \forall x \in S)$ where g^{α} is the master secret key held by the authority, g, h are in the public parameter, U_x is the public parameter corresponding to attribute x, and t is a random exponent chosen by the authority for the corresponding user of SK. However, the idea of using t as the personalized information for the key owner to achieve traceability is infeasible. A malicious owner of such a decryption key can leak a randomized version of the decryption key, for example, $\mathsf{SK}' = (K' = K \cdot h^{t'}, L' = L \cdot g^{t'}, K'_x = K_x \cdot U_x^{t'}\ \forall x \in S')$ where t' is a random exponent chosen by the key owner and $S' \subseteq S$. The re-randomization technique will prevent most of the expressive (CP- and KP-) ABE systems and their variants from being traceable.

To achieve traceability, in each decryption key we need a "*fixed point*" so that the key owner is not able to re-randomize it when he intends to leak out his decryption key as a new key. In our construction, starting with a digital signature scheme and making use of its existential unforgeability property, we "inject" it into the CP-ABE scheme of [10]. The message to be signed by the signature scheme will act as the "fixed point" as well as the "identifier" of the key owner. Note that this injection is not a trivial addition of signature generation and verification process into the key-generation and decryption process of the underlying CP-ABE scheme, respectively. Trivially combing a signature scheme and a CP-ABE scheme will result in an *artificial* system, as the signing and verifying operations are not necessary for key-generation and decryption, and cause unnecessary computation cost (of signing and verifying).

Inspired by Boneh and Boyen's signature scheme [18][2], we construct a Traceable CP-ABE system. In our system, the decryption key is in the form of $\mathsf{SK} = (K = g^{\alpha/(a+c)}h^{t}, K' = c, L = g^{t}, L' = g^{at}, K_x = U_x^{(a+c)t}\ \forall x \in S)$ where c is used as the "identifier" of the user. In the concrete construction (described in Section IV), we will see that our construction injects the signature scheme into the CP-ABE scheme *naturally*, and there are not unnecessary/artificial operations. Finally, we *add* traceability into the CP-ABE system of [10] at a very low cost (two additional elements in decryption key and one additional element in ciphertext).

C. Related Work

Another branch of ABE research is the multi-authority ABE [19]–[23], where the attributes are managed and issued by multiple authorities. These ABE systems do not consider the problem of resisting malicious key delegation.

Black-Box Traceable ABE Systems. In our construction, we target to make decryption key leakage to be traceable in the white-box model, i.e., the decryption keys leaked/sold will be used by the buyers to perform decryption using the ABE decryption algorithm. In practice, a stronger traceability notion is called black-box traceability, which is analogous to the notion of black-box traitor tracing in broadcast encryption [24], [25]. In particular, given a decryption equipment (where the embedded decryption key or algorithm could be unknown or hidden), the buyers can use it to retrieve plaintexts from ciphertexts. A black-box traceable ABE should allow an authority to find out the identity of the malicious user (i.e., whose decryption keys are used to create this decryption equipment).

Although the motivation and concept of traitor tracing in broadcast encryption [24], [25] are similar to the traceability in ABE, the techniques are essentially different. In broadcast encryption, each user has a personalized *index*, and the encryptor can specify the indices of the target receivers. i.e., the ciphertexts are related to some indices, each of which identifies a user. This is fundamentally different from the settings of CP-ABE, where the encryptor can encrypt data with access policy over descriptive attributes, without knowing or specifying exactly who the receivers are, so that the ciphertexts do not need to include any identification information of the receivers. The problem of black-box traitor tracing in broadcast encryption has been studied for a long time, and the main technique in the existing traitor tracing systems is to feed the decryption equipment with *tracing ciphertexts* which are related to the suspicious key owners but indistinguishable from *normal ciphertexts*. For broadcast encryption, the indistinguishability between the *tracing ciphertexts* and the *normal ciphertexts* is natural and easy to achieve, because both are inherently related to some specific indices. For CP-ABE systems, however, this seems to be hard to achieve: the traceability requires a *tracing ciphertext* to be related to some suspicious users, while the flexible expressiveness allows the *normal ciphertext*' access policy not to contain any specific information that can be used to identify particular users. This difference in essence seems to be preventing the techniques in broadcast encryption from being applied to CP-ABE.

Although [15] and [14] made attempts to achieve black-box traceability in the settings of CP-ABE and KP-ABE respectively, a decryption equipment can be constructed to resist their tracing algorithms by distinguishing the *tracing ciphertexts* from *normal ciphertexts*, as shown in Appendix A. They can be regarded as examples of the difficulty of achieving black-box traceable ABE systems and we consider this as one of our future work.

D. Outline

The formal definition of the traceable CP-ABE and its security model are given in the next section. Then the backgrounds, including the access structures, the bilinear map, and the assumptions, are reviewed in Section III. The main construction and security analysis are given in Section IV, and some extensions are discussed in Section V. Finally, the paper is concluded in Section VI.

II. Traceable CP-ABE

A. Definition

A Traceable CP-ABE (T-CP-ABE) system is a CP-ABE system where the decryption key can be traced to the corresponding owner. Then, naturally we add users' identities to the conventional CP-ABE system and add a Trace algorithm as well. In particular, following the notations of the conventional

[2] Roughly speaking, in the signature scheme [18], the signature for a message m is $g^{1/(a+m)}$, where a is the secret key for signing and (g, g^{a}) is the public key for verification.

CP-ABE system [10], a T-CP-ABE system is composed of the following five algorithms:

- $\mathsf{Setup}(\lambda, U) \to (\mathsf{PK}, \mathsf{MSK})$. The setup algorithm takes as input a security parameter $\lambda \in \mathbb{N}$ and an attribute universe description U. It outputs a public parameter PK and a master secret key MSK. In addition, it initializes an identity table $T = \emptyset$.
- $\mathsf{Encrypt}(\mathsf{PK}, \mathbb{A}, M) \to CT$. The encryption algorithm takes as input the public parameter PK, an access policy $\mathbb{A}$ over U, and a message M. It will output a ciphertext CT such that only the users whose decryption keys satisfy $\mathbb{A}$ should be able to extract M. $\mathbb{A}$ is implicitly included in CT.
- $\mathsf{KeyGen}(\mathsf{MSK}, \mathsf{PK}, id, S) \to \mathsf{SK}_{id,S}$. The key generation algorithm takes as input the master secret key MSK, the public parameter PK, and an attribute set S for a user with identity id. It outputs a decryption key $\mathsf{SK}_{id,S}$, and puts id into T.
- $\mathsf{Decrypt}(\mathsf{PK}, CT, \mathsf{SK}_{id,S}) \to M$ or $\perp$. The decryption algorithm takes as input the public parameter PK, a ciphertext CT, and a decryption key $\mathsf{SK}_{id,S}$. If S satisfies the access policy of the ciphertext, the algorithm outputs a message M, otherwise it outputs $\perp$ indicating the failure of decryption.
- $\mathsf{Trace}(\mathsf{PK}, T, \mathsf{SK}) \to id$ or $\top$. The tracing algorithm takes as input the public parameter PK, the table T and a decryption key SK. The algorithm first verifies whether SK is *well-formed* to determine whether SK needs to be traced. If SK is well-formed, the algorithm outputs an identity id implying that SK is linked to id, otherwise it outputs a special symbol $\top$ implying that SK does not need to be traced. A decryption key SK is "well-formed" means that SK passes a "key sanity check" which guarantees that the decryption key can be used in the well-formed decryption process. Similar to the work of [16], [17], the key sanity check will be defined as a deterministic algorithm.

B. IND-CPA Security

The security model for T-CP-ABE is very similar to that of the conventional CP-ABE [10], except that each key query is associated with an explicit identity. The following is the standard semantic security game (ciphertext indistinguishability under chosen plaintext attacks: IND-CPA).

- **Setup**. The challenger runs the Setup algorithm and gives the public parameter PK to the attacker.
- **Phase 1**. The attacker queries the challenger for decryption keys corresponding to sets of attributes $(id_1, S_1), \ldots, (id_{q_1}, S_{q_1})$.
- **Challenge**. The attacker declares two equal length messages M_0 and M_1 and an access policy $\mathbb{A}^*$. The challenger flips a random coin $\beta \in \{0, 1\}$, and encrypts M_β under $\mathbb{A}^*$, producing CT^*. It gives CT^* to the attacker.
- **Phase 2**. The attacker queries the challenger for decryption keys corresponding to sets of attributes $(id_{q_1+1}, S_{q_1+1}), \ldots, (id_q, S_q)$.
- **Guess**. The attacker outputs a guess β' for β.

The attacker wins the game if $\beta' = \beta$ under the **restriction** that $\mathbb{A}^*$ cannot be satisfied by any of the queried attribute sets $S_1, \ldots, S_q$. The advantage of an attacker in this game is defined to be $|\Pr[\beta' = \beta] - 1/2|$.

We note that the model can easily be extended to handle chosen-ciphertext attacks by allowing for decryption queries in Phase 1 and Phase 2.

Definition 1: A traceable ciphertext-policy attribute-based encryption system is fully secure if all polynomial time attackers have at most negligible advantage in this security game.

C. Traceability

We now give the traceability definition for T-CP-ABE. This is described by a security game between a challenger and an attacker as follows:

- **Setup**. The challenger runs the Setup algorithm and gives the public parameter PK to the attacker.
- **Key Query**. The attacker queries the challenger for decryption keys corresponding to sets of attributes $(id_1, S_1), \ldots, (id_q, S_q)$.
- **Key Forgery**. The attacker outputs a decryption key SK_*.

The attacker wins the game if $\mathsf{Trace}(\mathsf{PK}, T, \mathsf{SK}_*) \neq \top$ (i.e., SK_* is well-formed) **and** $\mathsf{Trace}(\mathsf{PK}, T, \mathsf{SK}_*) \notin \{id_1, \ldots, id_q\}$. The advantage of an attacker in this game is defined to be $\Pr[\mathsf{Trace}(\mathsf{PK}, T, \mathsf{SK}_*) \notin \{\top\} \cup \{id_1, \ldots, id_q\}]$.

Definition 2: A traceable ciphertext-policy attribute-based encryption system is fully traceable if all polynomial time attackers have at most negligible advantage in this game.

III. Background

A. Access Policy

Definition 3 (Access Structure [26]): Let $\{P_1, P_2, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1, P_2, \ldots, P_n\}}$ is monotone if $\forall B, C$: if $B \in \mathbb{A}$ *and* $B \subseteq C$ then $C \in \mathbb{A}$. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) $\mathbb{A}$ of non-empty subsets of $\{P_1, P_2, \ldots, P_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{P_1, P_2, \ldots, P_n\}} \setminus \{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.

In ABE, the role of the parties is taken by the attributes. Thus, the access structure $\mathbb{A}$ contains the authorized sets of attributes. As of previous work in ABE, we focus on monotone access structure in this paper. It is shown in [26] that any monotone access structure can be realized by a linear secret sharing scheme. Here we use the definition from [9], [26].

Definition 4 (Linear Secret-Sharing Schemes(LSSS) [9]): A secret-sharing scheme Π over a set of parties $\mathbb{P}$ is called linear (over $\mathbb{Z}_p$) if

1) The shares for each party form a vector over $\mathbb{Z}_p$.
2) There exists a matrix A called the share-generating matrix for Π. The matrix A has m rows and n columns. For $i = 1, \ldots, m$, the i-th row A_i of A is labeled by a party $\rho(i)$ (ρ is a function from $\{1, \ldots, m\}$ to $\mathbb{P}$). 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 $A\vec{v}$ is the vector of m shares of the secret s according to Π. The share $\lambda_i = (A\vec{v})_i$, i.e., the inner product $A_i \cdot \vec{v}$, belongs to party $\rho(i)$.

It is shown in [26] that every linear secret-sharing scheme according to the above definition also enjoys the linear reconstruction property, defined as follows: Suppose that Π is an LSSS for the access structure $\mathbb{A}$. Let $S \in \mathbb{A}$ be any authorized set, and let $I \subset \{1, 2, \ldots, m\}$ be defined as $I = \{i : \rho(i) \in S\}$. Then, there exist constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that if $\{\lambda_i\}$ are valid shares of any secret s according to Π, then $\sum_{i \in I} \omega_i \lambda_i = s$. Furthermore, it is shown in [26] that these constants $\{\omega_i\}$ can be found in time polynomial in the size of the share-generating matrix A. On the other hand, for any unauthorized set, no such constants exist. In this paper, we use an LSSS matrix (A, ρ) to express an access policy associated to a ciphertext.

B. Assumptions

1) Prime Order Bilinear Group and Assumptions*:* Let G and G_T be two (multiplicative) cyclic groups of prime order p, g be a generator of G, and $e : G \times G \rightarrow G_T$ be a bilinear map such that: (1) (Bilinear) $\forall\ g, h \in G, a, b \in \mathbb{Z}_p$, we have $e(g^a, h^b) = e(g, h)^{ab}$, (2) (Non-Degenerate) $e(g, g) \neq 1$. We say that G is a bilinear group if the group operations in G can be computed efficiently and there exists a group G_T and an efficiently computable bilinear map $e : G \times G \rightarrow G_T$ as above. Note that $e(\cdot, \cdot)$ is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

Boneh and Boyen introduced the l-Strong Diffie-Hellman (l-SDH) assumption [18] to build their short signature scheme without random oracles, and also introduced a weaker variant, the l-Bilinear Diffie-Hellman Inversion (l-BDHI) assumption [27], to construct an efficient selective identity secure IBE system without random oracles. Let G be a bilinear group of prime order p and g be a generator of G, given a $(l+1)$-tuple $(g, g^x, g^{x^2}, \ldots, g^{x^l})$ as input where $x \in \mathbb{Z}_p^*$ is randomly chosen, the l-SDH assumption states that there is no probabilistic polynomial time (PPT) algorithm that can output $(c, g^{1/(x+c)}) \in \mathbb{Z}_p^* \times G$ with non-negligible probability, and the l-BDHI assumption states that there is no PPT algorithm that can output $e(g, g)^{1/x}$ with non-negligible probability.

Note that in the l-SDH assumption, the c of the output $(c, g^{1/(x+c)})$ is not allowed to be zero. However, if the output is $(c = 0, g^{1/(x+c)} = g^{1/x})$, the output can be used to solve the l-BDHI problem. Similar to [16], we use the following definition of l-SDH assumption where c is allowed to be zero.

*Assumption 1 (**l-SDH**):* Let G be a bilinear group of prime order p and g be a generator of G, the l-Strong Diffie-Hellman (l-SDH) problem in G is defined as follows: given a $(l+1)$-tuple $(g, g^x, g^{x^2}, \ldots, g^{x^l})$ as input, output a pair $(c, g^{1/(x+c)}) \in \mathbb{Z}_p \times G$. An algorithm $\mathcal{A}$ has advantage ϵ in solving l-SDH in G if

$$\Pr\left[\mathcal{A}\left(g, g^x, g^{x^2}, \ldots, g^{x^l}\right) = \left(c, g^{\frac{1}{x+c}}\right)\right] \geq \epsilon$$

where the probability is over the random choice of x in $\mathbb{Z}_p^*$ and the random bits consumed by $\mathcal{A}$.

Definition 5: We say that (l, t, ϵ)-SDH assumption holds in G if no t-time algorithm has advantage at least ϵ in solving the l-SDH problem in G.

2) Composite Order Bilinear Group and Assumptions*:* Composite order bilinear groups are first introduced in [28] and then are widely used in IBE and ABE systems. Let $\mathcal{G}$ be the group generator, which takes a security parameter λ and outputs $(p_1, p_2, p_3, G, G_T, e)$ 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) (Bilinear) $\forall g, h \in G, a, b \in \mathbb{Z}_N, e(g^a, h^b) = e(g, h)^{ab}$, (2) (Non-Degenerate) $\exists g \in G$ such that $e(g, g)$ has order N in G_T. 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. 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 for any $h_i \in G_{p_i}$ and $h_j \in G_{p_j}$ where $i \neq j$, $e(h_i, h_j) = 1$.

For an element $T \in G$, T can (uniquely) be written as the product of an element of G_{p_1}, an element of G_{p_2}, and an element of G_{p_3}, and they are referred to as the "G_{p_1} part of T", "G_{p_2} part of T" and "G_{p_3} part of T", respectively. Let $G_{p_1 p_2}$ and $G_{p_1 p_3}$ be the subgroups of order $p_1 p_2$ and $p_1 p_3$ in G, respectively. Similarly, an element in $G_{p_1 p_2}$ can be written as the product of an element of G_{p_1} and an element of G_{p_2}, and an element in $G_{p_1 p_3}$ can be written as the product of an element of G_{p_1} and an element of G_{p_3}.

Assumption 2 (Subgroup Decision Problem for 3 Primes): [29] Given a group generator $\mathcal{G}$, define the following distribution:

$$\begin{aligned}
&\mathbb{G} = (N = p_1 p_2 p_3, G, G_T, e) \xleftarrow{R} \mathcal{G},\\
&g \xleftarrow{R} G_{p_1}, X_3 \xleftarrow{R} G_{p_3},\\
&D = (\mathbb{G}, g, X_3),\\
&T_1 \xleftarrow{R} G_{p_1 p_2}, T_2 \xleftarrow{R} G_{p_1}.
\end{aligned}$$

The advantage of an algorithm $\mathcal{A}$ in breaking Assumption 2 is: $Adv1_{\mathcal{G},\mathcal{A}} := |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|$.

Definition 6: We say that $\mathcal{G}$ satisfies Assumption 2 if $Adv1_{\mathcal{G},\mathcal{A}}$ is a negligible function of λ for any polynomial time algorithm $\mathcal{A}$.

Assumption 3: [29] Given a group generator $\mathcal{G}$, define the following distribution:

$$\begin{aligned}
&\mathbb{G} = (N = p_1 p_2 p_3, G, G_T, e) \xleftarrow{R} \mathcal{G},\\
&g, X_1 \xleftarrow{R} G_{p_1}, X_2, Y_2 \xleftarrow{R} G_{p_2}, X_3, Y_3 \xleftarrow{R} G_{p_3},\\
&D = (\mathbb{G}, g, X_1 X_2, X_3, Y_2 Y_3),\\
&T_1 \xleftarrow{R} G, T_2 \xleftarrow{R} G_{p_1 p_3}.
\end{aligned}$$

The advantage of an algorithm $\mathcal{A}$ in breaking Assumption 3 is:

$$Adv2_{\mathcal{G},\mathcal{A}} := |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|.$$

Definition 7: We say that $\mathcal{G}$ satisfies Assumption 3 if $Adv2_{\mathcal{G},\mathcal{A}}$ is a negligible function of λ for any polynomial time algorithm $\mathcal{A}$.

Assumption 4: [29] Given a group generator $\mathcal{G}$, define the following distribution:

$$\begin{aligned}
&\mathbb{G} = (N = p_1 p_2 p_3, G, G_T, e) \xleftarrow{R} \mathcal{G}, \alpha, s \xleftarrow{R} \mathbb{Z}_N,\\
&g \xleftarrow{R} G_{p_1}, X_2, Y_2, Z_2 \xleftarrow{R} G_{p_2}, X_3 \xleftarrow{R} G_{p_3},\\
&D = (\mathbb{G}, g, g^{\alpha} X_2, X_3, g^s Y_2, Z_2),\\
&T_1 = e(g, g)^{\alpha s}, T_2 \xleftarrow{R} G_T.
\end{aligned}$$

The advantage of an algorithm $\mathcal{A}$ in breaking Assumption 4 is:

$$Adv3_{\mathcal{G},\mathcal{A}} := |\Pr[\mathcal{A}(D,T_1)=1] - \Pr[\mathcal{A}(D,T_2)=1]|.$$

Definition 8: We say that $\mathcal{G}$ satisfies Assumption 4 if $Adv3_{\mathcal{G},\mathcal{A}}$ is a negligible function of λ for any polynomial time algorithm $\mathcal{A}$.

We will reduce the IND-CPA security of our T-CP-ABE scheme to that of the CP-ABE scheme in [10], which is based on Assumptions 2, 3 and 4. Here we would like to point out that: (1) all the three assumptions imply the hardness of factoring N; (2) the subgroup G_{p_i} is a bilinear group of order p_i, consequently the assumptions over bilinear group of prime order, such as l-SDH, will hold over G_{p_i}.

IV. Our Traceable CP-ABE System

A. Construction

- $\mathsf{Setup}(\lambda, U) \rightarrow (\mathsf{PK}, \mathsf{MSK})$. The algorithm runs the group generator $\mathcal{G}$ with λ as input and obtains $(p_1, p_2, p_3, G, G_T, e)$, where p_1, p_2, p_3 are distinct primes, G and G_T are cyclic groups of order $N = p_1p_2p_3$, and $e : G \times G \rightarrow G_T$ is a bilinear map. Let G_{p_i} be the subgroup of order p_i in G, and $g \in G_{p_1}, X_3 \in G_{p_3}$ be the generators of G_{p_1} and G_{p_3} respectively. The algorithm randomly chooses $\alpha, a \in \mathbb{Z}_N$ and $h \in G_{p_1}$, and for each $x \in U$, it randomly chooses $u_x \in \mathbb{Z}_N$. The public parameter PK is set to

$$\mathsf{PK} = (N, h, g, g^a, e(g,g)^\alpha, \quad \{U_x = g^{u_x}\}_{x \in U}).$$

The master secret key MSK is set to

$$\mathsf{MSK} = (\alpha, a, X_3).$$

The table T is initialized to be empty.

- $\mathsf{Encrypt}(\mathsf{PK}, (A, \rho), M) \rightarrow CT$. A is an $m \times n$ matrix and ρ maps each row A_i of A to an attribute $\rho(i)$. It is required that ρ would not map two different rows to the same attribute. The algorithm chooses a random vector $\vec{v} = (s, v_2, \ldots, v_n) \in \mathbb{Z}_N^n$. For each row A_i of A, it computes the inner product $\lambda_i = A_i \cdot \vec{v}$ and randomly picks $r_i \in \mathbb{Z}_N$. The ciphertext is set to

$$CT = \Big\langle C = M \cdot e(g,g)^{\alpha s}, C_0 = g^s, C_0' = g^{as}, \\ \Big\{C_i = h^{\lambda_i} U_{\rho(i)}^{-r_i}, C_i' = g^{r_i}\Big\}_{i=1}^{m}, \quad (A, \rho)\Big\rangle.$$

- $\mathsf{KeyGen}(\mathsf{MSK}, \mathsf{PK}, id, S) \rightarrow \mathsf{SK}_{id,S}$. The algorithm randomly chooses $c \in \mathbb{Z}_N^*$, $t \in \mathbb{Z}_N$, $R, R_0, R_0' \in G_{p_3}$, and for each $x \in S$, it randomly picks $R_x \in G_{p_3}$. The decryption key is set to

$$\mathsf{SK}_{id,S} = \Bigg(K = g^{\frac{\alpha}{a+c}} h^t R, K' = c, L = g^t R_0, L' = g^{at} R_0', \\ \Big\{K_x = U_x^{(a+c)t} R_x\Big\}_{x \in S}\Bigg).$$

Here $1/(a+c)$ is computed modulo N. In the unlikely events that $gcd(a+c, N) \neq 1$ or c has been in T, the algorithm repeats the above again using another randomly chosen value $c \in \mathbb{Z}_N^*$.

The algorithm puts tuple (c, id) into T.

- $\mathsf{Decrypt}(\mathsf{PK}, CT, \mathsf{SK}_{id,S}) \rightarrow M$ or $\perp$. The algorithm parses CT to $CT = \langle C, C_0, C_0', \{C_i, C_i'\}_{i=1}^m, (A, \rho)\rangle$ and $\mathsf{SK}_{id,S}$ to $\mathsf{SK}_{id,S} = (K, K', L, L', \{K_x\}_{x \in S})$. If the attribute set S cannot satisfy the access structure (A, ρ) of CT, the algorithm outputs $\perp$. Otherwise, the algorithm computes constants $\{\omega_i \in \mathbb{Z}_N\}$ such that $\sum_{\rho(i) \in S} \omega_i A_i = (1, 0, \ldots, 0)$, then computes:

$$D = \prod_{\rho(i) \in S} \left(e(L^{K'} L', C_i) \cdot e\left(K_{\rho(i)}, C_i'\right)\right)^{\omega_i} \\ = \prod_{\rho(i) \in S} e(g,h)^{(a+c)t\omega_i\lambda_i} = e(g,h)^{(a+c)ts},$$

$$E = e\left(K, C_0^{K'} C_0'\right) = e\left(g^{\frac{\alpha}{a+c}} h^t R, g^{(a+c)s}\right) \\ = e(g,g)^{\alpha s} e(g,h)^{(a+c)ts}.$$

Then M is recovered by $C \cdot D/E$.

- $\mathsf{Trace}(\mathsf{PK}, T, \mathsf{SK}) \rightarrow id$ or $\top$. If SK is in the form of $\mathsf{SK} = (K, K', L, L', \{K_x\}_{x \in S})$ and satisfies all of the following four checks, it is a well-formed decryption key whose decryption privilege is described by attribute set $S_w = \{x \,|\, x \in S \wedge e(U_x, L^{K'} \cdot L') = e(g, K_x) \neq 1\}$, otherwise it is not well-formed and the algorithm will output $\top$. If SK is well-formed, the algorithm will search K' in T: if K' can be found in T, the algorithm outputs the corresponding id, otherwise it outputs a special identity $id_\emptyset$ which never appears in T.

Key Sanity Check:

$$K' \in \mathbb{Z}_N, K, L, L', K_x \in G, \tag{1}$$

$$e(g, L') = e(g^a, L) \neq 1, \tag{2}$$

$$e(g^a \cdot g^{K'}, K) = e(g,g)^\alpha \cdot e(L^{K'} \cdot L', h) \neq 1, \tag{3}$$

$$\exists x \in S, s.t.\ e(U_x, L^{K'} \cdot L') = e(g, K_x) \neq 1. \tag{4}$$

Remarks: (1) A decryption key passing the sanity check can decrypt the ciphertexts with policies satisfied by S_w. (2) Any decryption key generated by the KeyGen algorithm can pass the sanity check with $S_w = S$, but a decryption key passing the sanity check may not be generated by the KeyGen algorithm: The check (1) requires $(K' \in \mathbb{Z}_N, K, L, L', K_x \in G)$ rather than $(K' \in \mathbb{Z}_N^*, K, L, L', K_x \in G_{p_1p_3})$. e.g., $(K = g^{\alpha/a} h^t R, K' = 0, L = g^t R_0, L' = g^{at} R_0', \{K_x = U_x^{at} R_x\}_{x \in S})$ is well-formed and can decrypt the ciphertexts with policies satisfied by S, although KeyGen algorithm never generates such a key.

The above scheme tries to add traceability property to the CP-ABE scheme of [10] which we will briefly review later. We should point out that

1) In the above scheme, it is required that an attribute appears at most once in an LSSS matrix (A, ρ). This restriction is inherited from the underlying CP-ABE scheme, where the security proof requires ρ is injective. Similar to [10] we call the above system as a One-Use system, and we can use the encoding technique in [10] to expand the One-Use system to a Multi-Use system, where there is not such a restriction.

2) The public parameter size is linear in the size of the attribute universe. Similarly, with the technique mentioned in [10], we can achieve a large universe construction.

B. IND-CPA Security

Although we can present a proof which is directly based on the Assumptions 2, 3 and 4 as [10] does, for simplicity, in this part we will reduce the security of our T-CP-ABE scheme to that of the scheme in [10]. We denote the CP-ABE scheme [10] as Σ_{cpabe}, and our scheme as Σ_{tcpabe}.

1) Brief Review of the Underlying CP–ABE System:

The construction of CP-ABE scheme $\boldsymbol{\Sigma_{cpabe}}$ [10]:

- $\mathsf{Setup}(\lambda, U) \rightarrow (\mathsf{PK}, \mathsf{MSK})$. Let G be a bilinear group of order $N = p_1p_2p_3$ (3 distinct primes), G_{p_i} be the subgroup of order p_i in G, and $g \in G_{p_1}, X_3 \in G_{p_3}$ be the generators of G_{p_1} and G_{p_3} respectively. The algorithm randomly chooses $\alpha, \beta \in \mathbb{Z}_N$, and for each $x \in U$, it randomly chooses $u_x \in \mathbb{Z}_N$. The public parameter is set to $\mathsf{PK} = (N, X_3, g, g^\beta, e(g,g)^\alpha, \{U_x = g^{u_x}\}_{x \in U})$, and the master secret key is set to $\mathsf{MSK} = (\alpha)$.
- $\mathsf{Encrypt}(\mathsf{PK}, (A, \rho), M) \rightarrow CT$. A is an $m \times n$ matrix and ρ maps each row A_i of A to an attribute $\rho(i)$. The algorithm chooses a random vector $\vec{v} = (s, v_2, \ldots, v_n) \in \mathbb{Z}_N^n$. For each row A_i of A, it computes the inner product $\lambda_i = A_i \cdot \vec{v}$ and randomly picks $r_i \in \mathbb{Z}_N$. The ciphertext is set to
$$CT = \Big\langle C = M \cdot e(g,g)^{\alpha s}, C_0 = g^s, \\ \Big\{C_i = g^{\beta\lambda_i} U_{\rho(i)}^{-r_i}, C'_i = g^{r_i}\Big\}_{i=1}^{m}, \quad (A, \rho)\Big\rangle .$$
- $\mathsf{KeyGen}(\mathsf{MSK}, \mathsf{PK}, S) \rightarrow \mathsf{SK}$. The algorithm randomly chooses $t \in \mathbb{Z}_N, R, R' \in G_{p_3}$, and for each $x \in S$ it randomly picks $R_x \in G_{p_3}$. The decryption key is set to
$$\mathsf{SK} = \Big(K = g^\alpha g^{\beta t} R, L = g^t R', \quad \{K_x = U_x^t R_x\}_{x \in S}\Big) .$$
- $\mathsf{Decrypt}(\mathsf{PK}, CT, \mathsf{SK}) \rightarrow M$. The algorithm computes constants $\{\omega_i \in \mathbb{Z}_N\}$ such that $\sum_{\rho(i) \in S} \omega_i A_i = (1, 0, \ldots, 0)$, and computes
$$\prod_{\rho(i) \in S} \big(e(L, C_i) e(K_{\rho(i)}, C'_i)\big)^{\omega_i} = e(g,g)^{\beta t s} .$$

Then M is recovered by $C \cdot e(g,g)^{\beta t s} / e(C_0, K)$.

Remarks: The above scheme is slightly different from the original one in [10] in that X_3 is in the public parameter rather than in the master secret key. Observe the proof detail of the original one in [10], we note that the above scheme can be proved with almost same proof, because the simulator is given X_3 explicitly in all the three underlying assumptions so that it can give X_3 to the adversary to achieve the proof of the above scheme. As X_3 is not used by the encryptors, the original scheme puts it in the master secret key.

The security of CP-ABE scheme $\boldsymbol{\Sigma_{cpabe}}$:

The security model of the underlying system in [10] is almost same with ours in Section II.B, except that the identity is not included. The details are referred to [10].

Lemma 1: [10] If Assumptions 2, 3, and 4 hold, then the CP-ABE scheme Σ_{cpabe} is secure.

2) IND–CPA Security of Our T–CP–ABE Scheme:

Lemma 2: If the CP-ABE scheme Σ_{cpabe} is secure in the security game of [10], then our T-CP-ABE scheme Σ_{tcpabe} is secure in the security game of Section II.B.

Proof: Suppose there is a PPT adversary $\mathcal{A}$ that can break our T-CP-ABE scheme Σ_{tcpabe} with advantage $Adv_{\mathcal{A}}\Sigma_{tcpabe}$, we construct a PPT algorithm $\mathcal{B}$ to break the underlying CP-ABE scheme Σ_{cpabe} with advantage $Adv_{\mathcal{B}}\Sigma_{cpabe}$, which equals to $Adv_{\mathcal{A}}\Sigma_{tcpabe}$.

- **Setup**. Σ_{cpabe} gives $\mathcal{B}$ the public parameter $\mathsf{PK}_{cpabe} = (N, X_3, g, g^\beta, e(g,g)^\alpha, \{U_x = g^{u_x}\}_{x \in U})$. $\mathcal{B}$ randomly chooses $a \in \mathbb{Z}_N$, then gives $\mathcal{A}$ public parameter $\mathsf{PK} = (N, h = g^\beta, g, g^a, e(g,g)^\alpha, \{U_x = g^{u_x}\}_{x \in U})$, and initializes the table $T = \emptyset$.
- **Phase 1**. When $\mathcal{A}$ submits (id, S) to $\mathcal{B}$ to request a decryption key, $\mathcal{B}$ submits S to Σ_{cpabe} and obtains the corresponding decryption key in the form of
$$\mathsf{SK} = \left(\bar{K} = g^\alpha g^{\beta\bar{t}} R, \bar{L} = g^{\bar{t}} R', \quad \Big\{\bar{K}_x = U_x^{\bar{t}} R_x\Big\}_{x \in S}\right) .$$
$\mathcal{B}$ randomly chooses $c \in \mathbb{Z}_N^*$ and computes $1/(a+c)$ modulo N. In the unlikely events that $gcd(a+c, N) \neq 1$ or c has been in T, $\mathcal{B}$ repeats it again using another randomly chosen value $c \in \mathbb{Z}_N^*$. Implicitly setting $t = \bar{t}/(a+c)$ and $K' = c$, $\mathcal{B}$ randomly chooses $R^{''} \in G_{p_3}$ by using X_3, then computes
$$K = (\bar{K})^{\frac{1}{a+c}} = (g^\alpha g^{\beta\bar{t}} R)^{\frac{1}{a+c}} = g^{\frac{\alpha}{a+c}} h^t R^{\frac{1}{a+c}}, \\ L = (\bar{L})^{\frac{1}{a+c}} = (g^{\bar{t}} R')^{\frac{1}{a+c}} = g^t R'^{\frac{1}{a+c}}, \\ L' = (\bar{L})^{\frac{a}{a+c}} R^{''} = (g^{\bar{t}} R')^{\frac{a}{a+c}} R^{''} = g^{a \cdot t} R'^{\frac{a}{a+c}} R^{''}, \\ \Big\{K_x = \bar{K}_x = U_x^{\bar{t}} R_x = U_x^{(a+c)t} R_x\Big\}_{x \in S} .$$
$\mathcal{B}$ gives $\mathcal{A}$ the decryption key $\mathsf{SK}_{id,S} = (K, K', L, L', \{K_x\}_{x \in S})$, and puts tuple (c, id) into T.
Note that $R^{''}$ makes the G_{p_3} part of L' uncorrelated to the G_{p_3} part of L, this is why our simulator needs X_3.
- **Challenge**. $\mathcal{A}$ submits to $\mathcal{B}$ an LSSS matrix (A^*, ρ) and two equal length messages M_0, M_1.
$\mathcal{B}$ submits $((A^*, \rho), M_0, M_1)$ to Σ_{cpabe}, and obtains the challenge ciphertext in the form of
$$\bar{C}T = \Big\langle \bar{C} = M_b \cdot e(g,g)^{\alpha s}, \bar{C}_0 = g^s, \\ \Big\{\bar{C}_i = g^{\beta\lambda_i} U_{\rho(i)}^{-r_i}, \bar{C}'_i = g^{r_i}\Big\}_{i=1}^{m}, \quad (A^*, \rho)\Big\rangle .$$
$\mathcal{B}$ gives $\mathcal{A}$ the challenge ciphertext as
$$CT = \Big\langle C = \bar{C} = M_b \cdot e(g,g)^{\alpha s}, \\ C_0 = \bar{C}_0 = g^s, C'_0 = \bar{C}_0^a = g^{as}, \\ \Big\{C_i = \bar{C}_i = h^{\lambda_i} U_{\rho(i)}^{-r_i}, C'_i = \bar{C}'_i = g^{r_i}\Big\}_{i=1}^{m}, \quad (A^*, \rho)\Big\rangle .$$
- **Phase 2**. Same with Phase 1.
- **Guess**. $\mathcal{A}$ gives $\mathcal{B}$ a b'. $\mathcal{B}$ gives b' to Σ_{cpabe}.

Note that the distributions of the public parameter, decryption keys and challenge ciphertext are same as the real scheme, we have $Adv_{\mathcal{B}}\Sigma_{cpabe} = Adv_{\mathcal{A}}\Sigma_{tcpabe}$. ■

Theorem 1: If Assumptions 2, 3, and 4 hold, then our T-CP-ABE scheme Σ_{tcpabe} is secure.

Proof: It follows from Lemma 1 and Lemma 2. ■

CCA Security. While we proved our T-CP-ABE scheme secure under chosen plaintext attacks, it is not difficult to modify it slightly and apply the methods of Canetti, Halevi, and Katz [30] for security against chosen ciphertext attacks.

C. Traceability

In this part, we prove the traceability of our T-CP-ABE scheme based on Assumption 3 and l-SDH assumption. The proof uses a similar method to the proof of Lemma 1 of Boneh and Boyen [18].

Theorem 2: If Assumption 3 and l-SDH assumption hold, then our T-CP-ABE scheme Σ_{tcpabe} is fully traceable provided that $q < l$.

Proof: Suppose there is a PPT adversary $\mathcal{A}$ that wins the traceability game with non-negligible advantage ϵ after making q key queries, without loss of generality, assuming $l = q + 1$, we construct a PPT algorithm $\mathcal{B}$ that can break Assumption 3 or l-SDH assumption with non-negligible advantage.

$\mathcal{B}$ is given an instance of Assumption 3 problem and an instance of l-SDH problem as follows. Note that the two instances are independent from each other, although we use the same symbols.

- $\mathcal{B}$ is given an instance of Assumption 3 problem: Let G be a bilinear group of order $N = p_1p_2p_3$ (3 distinct primes), G_{p_i} be the subgroup of order p_i in G $(1 \leq i \leq 3)$, $e : G \times G \rightarrow G_T$ be a bilinear map, $\bar{g}, X_1 \in G_{p_1}$, $X_2, Y_2 \in G_{p_2}$, and $X_3, Y_3 \in G_{p_3}$. $b \in \{0,1\}$, and $T \in G$ if $b = 0$, $T \in G_{p_1p_3}$ if $b = 1$. $\mathcal{B}$ is given $\mathsf{IN}_{\mathsf{A3}} = (N, G, G_T, e, \bar{g}, X_1X_2, X_3, Y_2Y_3, T)$.
- $\mathcal{B}$ is given an instance of l-SDH problem: Let G be a bilinear group of order $N = p_1p_2p_3$ (3 distinct primes), G_{p_i} be the subgroup of order p_i in G $(1 \leq i \leq 3)$, $e : G \times G \rightarrow G_T$ be a bilinear map, $\bar{g} \in G_{p_1}$ and $a \in \mathbb{Z}_{p_1}^*$. $\mathcal{B}$ is given an instance $(\bar{g}, \bar{g}^a, \ldots, \bar{g}^{a^l})$ of l-SDH problem in the subgroup G_{p_1}. Also $\mathcal{B}$ is given the factors p_1, p_2, and p_3[3]. i.e., $\mathcal{B}$ is given $\mathsf{IN}_{\mathsf{SDH}} = (N, G, G_T, e, \bar{g}, \bar{g}^a, \ldots, \bar{g}^{a^l}, p_1, p_2, p_3)$.

$\mathcal{B}$'s goal is to output a bit $b' \in \{0,1\}$ to determine $T \in G$ or $T \in G_{p_1p_3}$ for solving the Assumption 3 problem, and a pair $(c_r, w_r) \in \mathbb{Z}_{p_1} \times G_{p_1}$ satisfying $w_r = \bar{g}^{1/(a+c_r)}$ for solving the l-SDH problem. $\mathcal{B}$ will make use of $\mathcal{A}$ to break at least one of Assumption 3 and l-SDH assumption.

Before playing the traceability game with $\mathcal{A}$, $\mathcal{B}$ flips a random coin $\theta \in \{0,1\}$,

- if $\theta = 0$, taking $\mathsf{IN}_{\mathsf{A3}} = (N, G, G_T, e, \bar{g}, X_1X_2, X_3, Y_2Y_3, T)$ as input, $\mathcal{B}$ randomly chooses $\hat{a} \in \mathbb{Z}_N^*$, and computes $A_i = \bar{g}^{\hat{a}^i}$ for $i = 0, 1, \ldots, l$. Implicitly, unknown value $a = (\hat{a} \bmod p_1) \in \mathbb{Z}_{p_1}^*$ is randomly chosen and $A_i = \bar{g}^{\hat{a}^i} = \bar{g}^{a^i}$ for $i = 0, 1, \ldots, l$ are set.
- else, taking $\mathsf{IN}_{\mathsf{SDH}} = (N, G, G_T, e, \bar{g}, \bar{g}^a, \ldots, \bar{g}^{a^l}, p_1, p_2, p_3)$ as input, $\mathcal{B}$ sets $A_i = \bar{g}^{a^i}$ for $i = 0, 1, \ldots, l$, and chooses a generator X_3 of G_{p_3} (Note that $\mathcal{B}$ can choose such a X_3 since it knows the values of p_1, p_2, and p_3).

Then $\mathcal{B}$ takes $(N, G, G_T, e, X_3, A_i\ (i = 0, 1, \ldots, l))$ as input to interact with $\mathcal{A}$ in the traceability game as follows:

- **Setup**. $\mathcal{B}$ randomly chooses q distinct values $c_1, \ldots, c_q \in \mathbb{Z}_N^*$. 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 $\alpha_0, \alpha_1, \ldots, \alpha_q \in \mathbb{Z}_N$ are the coefficients of the polynomial $f(y)$. $\mathcal{B}$ computes
$$g \leftarrow \prod_{i=0}^{q}(A_i)^{\alpha_i} = \bar{g}^{f(a)} \in G_{p_1},$$
$$g^a \leftarrow \prod_{i=1}^{q+1}(A_i)^{\alpha_{i-1}} = \bar{g}^{f(a)\cdot a}.$$
$\mathcal{B}$ randomly chooses $\alpha, \beta \in \mathbb{Z}_N$, and for each $x \in U$ it randomly picks $u_x \in \mathbb{Z}_N$, then gives $\mathcal{A}$ the public parameter
$$\mathsf{PK} = \left(N, h = g^{\beta}, g, g^a, e(g,g)^{\alpha}, \quad \{U_x = g^{u_x}\}_{x \in U}\right).$$
- **Key Query**. $\mathcal{A}$ submits (id_i, S_i) to $\mathcal{B}$ to request a decryption key. Assume it is the i-th 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 $f_i(y) = \sum_{j=0}^{q-1} \beta_j y^j$. $\mathcal{B}$ computes
$$\sigma_i \leftarrow \prod_{j=0}^{q-1}(A_j)^{\beta_j} = \bar{g}^{f_i(a)} = \bar{g}^{f(a)/(a+c_i)} = g^{1/(a+c_i)}.$$
$\mathcal{B}$ randomly chooses $t \in \mathbb{Z}_N$, $R, R_0, R_0' \in G_{p_3}$, and for each $x \in S_i$, it randomly picks $R_x \in G_{p_3}$. $\mathcal{B}$ responds $\mathcal{A}$ with $\mathsf{SK}_{id_i, S_i} = (K, K', L, L', \quad \{K_x\}_{x \in S_i})$ where
$$K = (\sigma_i)^{\alpha} h^t R = g^{\alpha/(a+c_i)} h^t R, \quad K' = c_i,$$
$$L = g^t R_0, \quad L' = g^{at} R_0',$$
$$\left\{K_x = (g^a \cdot g^{c_i})^{u_x t} R_x = U_x^{(a+c_i)t} R_x\right\}_{x \in S_i}.$$
$\mathcal{B}$ puts tuple (c_i, id_i) into T.
- **Key Forgery**. $\mathcal{A}$ submits to $\mathcal{B}$ a decryption key SK_*.

Note that the distributions of $(N, G, G_T, e, X_3, A_i (i = 0, 1, \ldots, l))$ for $\theta = 0$ and $\theta = 1$ are same, and $\mathcal{B}$ uses only $(N, G, G_T, e, X_3, A_i\ (i = 0, 1, \ldots, l))$ as input to interact with $\mathcal{A}$. We have that the value of θ is information-theoretically hidden from $\mathcal{A}$. Also note that the distributions of PK and $\{\mathsf{SK}\}$ in the above game are same as in the real scheme.

Let $\mathcal{E}_{\mathcal{A}}$ denote the event that $\mathcal{A}$ wins the game, i.e., SK_* is in the form of $\mathsf{SK}_* = (K, K', L, L', \{K_x\}_{x \in S_*})$ and satisfies the checks of (1), (2), (3) and (4), and $K' \notin \{c_1, c_2, \ldots, c_q\}$.

If $\mathcal{E}_{\mathcal{A}}$ does not happen, $\mathcal{B}$ chooses a random $b' \in \{0,1\}$ and a random pair $(c_r, w_r) \in \mathbb{Z}_{p_1} \times G_{p_1}$ (of $\mathsf{IN}_{\mathsf{SDH}}$) as its solutions for the Assumption 3 problem and l-SDH problem respectively.

If $\mathcal{E}_{\mathcal{A}}$ happens, using long division $\mathcal{B}$ writes the polynomial f as $f(y) = \gamma(y)(y + K') + \gamma_{-1}$ for some polynomial $\gamma(y) = \sum_{i=0}^{q-1} \gamma_i y^i$ and some $\gamma_{-1} \in \mathbb{Z}_N$. We have $\gamma_{-1} \neq 0$, since $f(y) = \prod_{i=1}^{q}(y + c_i)$, $c_i \in \mathbb{Z}_N^*$ and $K' \notin \{c_1, c_2, \ldots, c_q\}$, as

[3] The situation is similar to that of the proof in [25] in the sense that the challenge is given in a subgroup of a composite order group and the factors are given to the simulator.

thus $y+K'$ does not divide $f(y)$. Then $\mathcal{B}$ computes the value of $gcd(\gamma_{-1}, N)$, and either $gcd(\gamma_{-1}, N) \neq 1$ or $gcd(\gamma_{-1}, N) = 1$ happens:

- **Case I:** $gcd(\gamma_{-1}, N) \neq 1$.

  If $\theta = 1$, it implies that $\mathcal{B}$ made use of $\mathsf{IN}_{\mathsf{SDH}}$ to interact with $\mathcal{A}$. As $gcd(\gamma_{-1}, N) \neq 1$ does not provide useful information to $\mathcal{B}$, $\mathcal{B}$ chooses a random $b' \in \{0,1\}$ and a random pair $(c_r, w_r) \in \mathbb{Z}_{p_1} \times G_{p_1}$ (of $\mathsf{IN}_{\mathsf{SDH}}$) as its outputs for the Assumption 3 problem and l-SDH problem respectively.

  If $\theta = 0$, it implies that $\mathcal{B}$ made use of $\mathsf{IN}_{\mathsf{A3}}$ to interact with $\mathcal{A}$. $\mathcal{B}$ chooses a random pair $(c_r, w_r) \in \mathbb{Z}_{p_1} \times G_{p_1}$ (of $\mathsf{IN}_{\mathsf{SDH}}$) as its output for the l-SDH problem, and determines the value of b' as follows:

  Using the value of $gcd(\gamma_{-1}, N) \neq 1$, $\mathcal{B}$ obtains two non-trivial factors $n, n' \in \mathbb{Z}_N$ such that $nn' = N$. We have that $(n, n') \in \{(p_1, p_2p_3), (p_2p_3, p_1), (p_2, p_1p_3), (p_1p_3, p_2), (p_3, p_1p_2), (p_1p_2, p_3)\}$.

  — If $\bar{g}^{n} = 1$ and $(Y_2Y_3)^{n'} = 1$, $\mathcal{B}$ knows that $n = p_1$ and $n' = p_2p_3$. Otherwise, if $\bar{g}^{n'} = 1$ and $(Y_2Y_3)^{n} = 1$, $\mathcal{B}$ knows that $n = p_2p_3$ and $n' = p_1$. i.e., $\mathcal{B}$ obtains the value of p_1.

  Then $\mathcal{B}$ computes the value of $e(T^{p_1}, X_1X_2)$. If $e(T^{p_1}, X_1X_2) = 1$, $\mathcal{B}$ sets $b' = 1$, otherwise sets $b' = 0$ (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 $X_3^{n} = 1$ and $(X_1X_2)^{n'} = 1$, $\mathcal{B}$ knows that $n = p_3$ and $n' = p_1p_2$. Otherwise, if $X_3^{n'} = 1$ and $(X_1X_2)^{n} = 1$, $\mathcal{B}$ knows that $n = p_1p_2$ and $n' = p_3$. i.e., $\mathcal{B}$ obtains the value of p_3.

  Then $\mathcal{B}$ computes the value of $e(T^{p_3}, Y_2Y_3)$. If $e(T^{p_3}, Y_2Y_3) = 1$, $\mathcal{B}$ sets $b' = 1$, otherwise sets $b' = 0$ (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 $X_3^{n} = 1$, $\mathcal{B}$ knows that $n = p_1p_3$ and $n' = p_2$. Otherwise, if $X_3^{n'} = 1$, $\mathcal{B}$ knows that $n = p_2$ and $n' = p_1p_3$. i.e., $\mathcal{B}$ obtains the value of p_2.

  Then $\mathcal{B}$ computes the value of $T^{p_1p_3}$. If $T^{p_1p_3} = 1$, $\mathcal{B}$ sets $b' = 1$, otherwise sets $b' = 0$ (since $T^{p_1p_3} \in G_{p_2}$ if $T \in G$ and $T^{p_1p_3} = 1$ if $T \in G_{p_1p_3}$).

- **Case II:** $gcd(\gamma_{-1}, N) = 1$.

  If $\theta = 0$, it implies that $\mathcal{B}$ made use of $\mathsf{IN}_{\mathsf{A3}}$ to interact with $\mathcal{A}$. As $gcd(\gamma_{-1}, N) = 1$ does not provide useful information to $\mathcal{B}$, $\mathcal{B}$ chooses a random $b' \in \{0,1\}$ and a random pair $(c_r, w_r) \in \mathbb{Z}_{p_1} \times G_{p_1}$ (of $\mathsf{IN}_{\mathsf{SDH}}$) as its outputs for the Assumption 3 problem and l-SDH problem respectively.

  If $\theta = 1$, it implies that $\mathcal{B}$ made use of $\mathsf{IN}_{\mathsf{SDH}}$ to interact with $\mathcal{A}$. $\mathcal{B}$ chooses a random $b' \in \{0,1\}$ as its output for the Assumption 3 problem, and computes a pair $(c_r, w_r) \in \mathbb{Z}_{p_1} \times G_{p_1}$ (of $\mathsf{IN}_{\mathsf{SDH}}$) as follows:

  Assuming $L = g^{t} L_2 L_3$ where $t \in \mathbb{Z}_N$, $L_2 \in G_{p_2}, L_3 \in G_{p_3}$ are unknown, we have

$$\begin{aligned} L' &= g^{at} L_2' L_3' \quad \text{from (2)}, \\ K &= g^{\frac{\alpha}{a+K'}} h^{t} K_2 K_3 \quad \text{from (2) and (3)}, \end{aligned}$$

  where $L_2', K_2 \in G_{p_2}$ and $L_3', K_3 \in G_{p_3}$.

  $\mathcal{B}$ computes $1/\gamma_{-1} \bmod N$ (since $gcd(\gamma_{-1}, N) = 1$), then computes

$$\begin{aligned} \sigma &\leftarrow \left((K/L^{\beta})^{p_2p_3}\right)^{(p_2p_3\alpha)^{-1} \bmod p_1} \\ &= g^{\frac{1}{a+K'}} = \bar{g}^{\gamma(a)} \bar{g}^{\frac{\gamma_{-1}}{a+K'}}, \\ w_r &\leftarrow \left(\sigma \cdot \prod_{i=0}^{q-1} A_i^{-\gamma_i}\right)^{1/\gamma_{-1}} \quad = \bar{g}^{\frac{1}{a+K'}} \in G_{p_1}, \\ c_r &\leftarrow K' \bmod p_1 \in \mathbb{Z}_{p_1}. \end{aligned}$$

  Note that (c_r, w_r) is a solution for the l-SDH problem, since $e(\bar{g}^{a} \cdot \bar{g}^{c_r}, w_r) = e(\bar{g}^{a} \cdot \bar{g}^{K'}, \bar{g}^{(1)/(a+K')}) = e(\bar{g}, \bar{g})$.

We now evaluate the advantages of $\mathcal{B}$ in breaking Assumption 3 and l-SDH assumption.

Note that when $\mathcal{B}$ chooses $b' \in \{0,1\}$ randomly, $b' = b$ happens with probability $1/2$. Also note that the b' output by $\mathcal{B}$ in the case of $(\mathcal{A}\,win \wedge gcd(\gamma_{-1}, N) \neq 1 \wedge \theta = 0)$ satisfies $b' = b$ with probability 1.

$\mathcal{B}$ solves the Assumption 3 problem with probability

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

Let $\mathcal{E}_{\mathsf{SDH}}(c_r, w_r)$ denote the event that (c_r, w_r) is a solution for the l-SDH problem, which can be verified by checking whether $e(\bar{g}^{a} \cdot \bar{g}^{c_r}, w_r) = e(\bar{g}, \bar{g})$ holds. Note that when $\mathcal{B}$ chooses (c_r, w_r) randomly, $\mathcal{E}_{\mathsf{SDH}}(c_r, w_r)$ happens with negligible probability, for simplicity, say zero. Also note that the (c_r, w_r) output by $\mathcal{B}$ in the case of $(\mathcal{A}\,win \wedge gcd(\gamma_{-1}, N) = 1 \wedge \theta = 1)$ satisfies $e(\bar{g}^{a} \cdot \bar{g}^{c_r}, w_r) = e(\bar{g}, \bar{g})$ with probability 1.

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

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

The advantage of $\mathcal{B}$ in breaking Assumption 3 is

$$
\begin{aligned}
Adv_{\mathcal{B},A3} &= \left| \Pr[b' = b] - \frac{1}{2} \right|\\
&= \frac{1}{4} \cdot \Pr[\mathcal{A}\,win \wedge gcd(\gamma_{-1}, N) \neq 1].
\end{aligned}
$$

The advantage of $\mathcal{B}$ in breaking l-SDH assumption is

$$
\begin{aligned}
Adv_{\mathcal{B},\mathsf{SDH}} &= \Pr[\mathcal{E}_{\mathsf{SDH}}(c_r, w_r)]\\
&= \frac{1}{2} \cdot \Pr[\mathcal{A}\,win \wedge gcd(\gamma_{-1}, N) = 1].
\end{aligned}
$$

Note that

$$
\begin{aligned}
&Adv_{\mathcal{B},A3} + Adv_{\mathcal{B},\mathsf{SDH}}\\
&\quad= \frac{1}{4} \cdot \Pr[\mathcal{A}\,win \wedge gcd(\gamma_{-1}, N) \neq 1]\\
&\qquad+ \frac{1}{2} \cdot \Pr[\mathcal{A}\,win \wedge gcd(\gamma_{-1}, N) = 1]\\
&\quad= \frac{1}{4} \cdot \Pr[\mathcal{A}\,win] + \frac{1}{4} \cdot \Pr[\mathcal{A}\,win \wedge gcd(\gamma_{-1}, N) = 1]\\
&\quad= \frac{\epsilon}{4} + \frac{1}{4} \cdot \Pr[\mathcal{A}\,win \wedge gcd(\gamma_{-1}, N) = 1] \geq \frac{\epsilon}{4}.
\end{aligned}
$$

According to Dirichlet's drawer principle we have that at least one of $Adv_{\mathcal{B},A3}$, $Adv_{\mathcal{B},\mathsf{SDH}}$ is $\geq \epsilon/8$. i.e., $\mathcal{B}$ can break Assumption 3 with advantage $\geq \epsilon/8$ or break l-SDH assumption with advantage $\geq \epsilon/8$. ■

V. Extensions

A. *Reducing Trust on Authority*

In the definition in Section II and the construction in Section IV, the authority is assumed to be completely trusted, so that it never maliciously issues decryption keys. This is the inherent key escrow problem of ABE system. We can reduce the trust on authority somewhat as follows.

We split the authority into two different authorities: a Central Authority (CA) and a Tracing Authority (TA), where CA is responsible for issuing decryption keys to users and TA is responsible for tracing. In the following extended construction, although CA can still decrypt all ciphertexts, neither CA nor TA can independently generate well-formed decryption keys. Note that this is orthogonal from the techniques of the existing multi-authority ABE systems.

- $\mathsf{GlobalSetup}(\lambda) \to (\mathsf{GPK})$. Let G be a bilinear group of order $N = p_1p_2p_3$ (3 distinct primes), and G_{p_i} be the subgroup of order p_i in G. The algorithm, run by a trusted party which is involved only at the setup phase, randomly chooses $g, h \in G_{p_1}$ and $X_3 \in G_{p_3}$. The global public parameter is set to $\mathsf{GPK} = (N, X_3, g, h)$.
- $\mathsf{TASetup}(\mathsf{GPK}) \to (\mathsf{TPK}, \mathsf{TMSK})$. The algorithm, run by TA, randomly chooses an exponent $a \in \mathbb{Z}_N$. The public parameter is set to $\mathsf{TPK} = (g^a)$, and the master secret key is set to $\mathsf{TMSK} = (a)$. The identity table T is initialized to $T = \emptyset$.
- $\mathsf{CASetup}(\mathsf{GPK}, U) \to (\mathsf{CPK}, \mathsf{CMSK})$. The algorithm, run by CA, randomly chooses an exponent $\alpha \in \mathbb{Z}_N$, and for each $x \in U$, it randomly picks $u_x \in \mathbb{Z}_N$. The public parameter is set to $\mathsf{CPK} = (e(g,g)^\alpha, \{U_x = g^{u_x}\}_{x \in U})$, and the master secret key is set to $\mathsf{CMSK} = (\alpha, \{u_x\}_{x \in U})$.
- $\mathsf{TKeyGen}(\mathsf{TMSK}, \mathsf{GPK}, \mathsf{TPK}, id) \to \sigma_{id}$. When a user submits his identity id to TA, TA runs the algorithm. The algorithm randomly chooses $c \in \mathbb{Z}_N^*$, and computes $\delta = g^{1/(a+c)}, \delta' = g^c$. Then TA gives the user $\sigma_{id} = (c, \delta, \delta')$. Here $1/(a + c)$ is computed modulo N. In the unlikely events that $gcd(a + c, N) \neq 1$ or c has been in T, the algorithm repeats the above again using another randomly chosen value $c \in \mathbb{Z}_N^*$. The algorithm puts (c, id) into T.
- $\mathsf{CKeyGen}(\mathsf{CMSK}, \mathsf{GPK}, \mathsf{TPK}, \mathsf{CPK}, (\delta, \delta'), S) \to \mathsf{CSK}_{id,S}$. When a user submits his (δ, δ') and his attribute set S to CA, CA runs the algorithm. The algorithm first verifies that $e(g^a \cdot \delta', \delta) = e(g, g)$ and (δ, δ') has not been submitted previously[4], if the checks do not hold, it outputs $\perp$ to imply the submitted (δ, δ') is invalid. Otherwise, it randomly chooses $t \in \mathbb{Z}_N$, $R, R_0, R'_0 \in G_{p_3}$, and for each $x \in S$ it randomly picks $R_x \in G_{p_3}$. Then it computes

$$
\begin{aligned}
\mathsf{CSK}_{id,S} = \Big(&K = \delta^\alpha h^t R, L = g^t R_0, L' = g^{at} R'_0,\\
&\left\{K_x = (g^a \cdot \delta')^{u_x t} R_x\right\}_{x \in S}\Big).
\end{aligned}
$$

  If a user obtains $\sigma_{id} = (c, \delta, \delta')$ from TA and the corresponding $\mathsf{CSK}_{id,S} = (K, L, L', \{K_x\}_{x \in S})$ from CA, his decryption key is $\mathsf{SK}_{id,S} = (K, K', L, L', \{K_x\}_{x \in S})$ where $K' = c$.
- $\mathsf{Encrypt}$, $\mathsf{Decrypt}$ and Trace. Let $\mathsf{PK} = \mathsf{GPK} \cup \mathsf{TPK} \cup \mathsf{CPK}$, the algorithms of $\mathsf{Encrypt}$, $\mathsf{Decrypt}$ and Trace are same with that of the main construction in Section IV.A.

In the above modified system, we assume that TA and CA are uncorrupted, as of the authorities in most existing ABE systems. Also, it is assumed that neither TA nor CA would collude with any user to maliciously generate well-formed decryption keys, as otherwise, they would have to risk themselves from being denounced by colluded users at some later time. The advantage of this modified system over the original one we proposed is that neither TA nor CA can *independently* generate

[4] This is to guarantee that each (δ, δ') can be used only once for requesting secret keys.

well-formed decryption keys, even if they intended to do that.

Against attacks defined in Sections II.B and II.C: The above modified system is as secure as the main construction in Section IV.

Against a malicious CA who attempts to independently generate well-formed decryption keys: CA is given GPK and TPK, then it outputs CPK. After given a series of pairs $(g^{1/(a+c_i)}, g^{c_i})$, CA outputs a decryption key SK_*. If SK_* is well-formed, a pair $(g^{\alpha/(a+c_*)}, c_*)$ can be induced from SK_*. If c_* is equal to some c_i, such a CA could be used to solve a Discrete-Logarithm-like problem, otherwise such a CA could be used to solve a l-SDH-like problem. Thus the security against malicious CA can be proved at least in the generic group model.

Against a malicious TA who attempts to independently generate well-formed decryption keys: TA is given GPK, then it outputs TPK, and is given CPK. Then TA outputs a decryption key SK_*. If SK_* is well-formed, a pair $(g^{\alpha/(a+c_*)}, c_*)$ can be induced from SK_*. It can be proved in the generic group model that TA cannot output a well-formed decryption key, because $e(g,g)^\alpha$ is the only element related to α that TA can obtain, but $e(g,g)^\alpha$ is not in the group G.

B. Removing the Identity Table

In the main construction, the Trace algorithm has to search K' in the identity table T to obtain the corresponding id. In this section, we extend our construction to eliminate the search cost so that it is more efficient in practice. Considering another signature scheme in [18][5], we explicitly embed the identity of user into the decryption key as follows (Here we just sketch the construction by giving the forms of the keys and ciphertext):

- Setup: $\mathsf{PK} = (N, h, g, g^a, g^b, e(g,g)^\alpha, \{U_x = g^{u_x}\}_{x\in U})$, $\mathsf{MSK} = (g^\alpha, a, b, X_3)$, where $\alpha, a, b, u_x (x \in U) \in \mathbb{Z}_N$, $g, h \in G_{p_1}$ and $X_3 \in G_{p_3}$.
- KeyGen: For identity $id \in \mathbb{Z}_N^*$ and attribute set $S \subseteq U$, exponents $r, t \in \mathbb{Z}_N$ and elements $R, R_0, R_0', R_x (x \in S) \in G_{p_3}$ are randomly chosen, and the decryption key is set to $\mathsf{SK}_{id,S} = (K, K', K'', L, L', \{K_x\}_{x\in S})$ where

$$K = g^{\frac{\alpha}{a+id+b\cdot r}} h^t R, K' = id, K'' = r,$$
$$L = g^t R_0, L' = g^{(a+b\cdot r)t} R_0',$$
$$\left\{K_x = U_x^{(a+id+b\cdot r)t} R_x\right\}_{x\in S}.$$

- $\mathsf{Encrypt}$: For a message M and an LSSS matrix (A,ρ), $\vec{v} = (s, v_2, \ldots, v_n) \in \mathbb{Z}_N^n$ and $r_i (1 \le i \le m) \in \mathbb{Z}_N$ are randomly chosen, and the ciphertext is

$$CT = \Big\langle C = M \cdot e(g,g)^{\alpha s}, C_0 = g^s, C_0' = g^{bs}, C_0'' = g^{as},$$
$$\left\{C_i = h^{A_i\cdot\vec{v}} U_{\rho(i)}^{-r_i}, C_i' = g^{r_i}\right\}_{i=1}^m, \quad (A,\rho)\Big\rangle.$$

- $\mathsf{Decrypt}$: The algorithm computes constants $\{\omega_i \in \mathbb{Z}_N\}$ such that $\sum_{\rho(i)\in S} \omega_i A_i = (1, 0, \ldots, 0)$, and computes

$$\begin{aligned} D &= \prod_{\rho(i)\in S} \left(e(L^{K'} L', C_i) \cdot e(K_{\rho(i)}, C_i')\right)^{\omega_i} \\ &= \prod_{\rho(i)\in S} e(g,h)^{(a+id+b\cdot r)t\omega_i\lambda_i} = e(g,h)^{(a+id+b\cdot r)ts}, \\ E &= e\left(K, C_0^{K'} (C_0')^{K''} C_0''\right) \\ &= e\left(g^{\frac{\alpha}{a+id+b\cdot r}} h^t R, g^{(a+id+b\cdot r)s}\right) \\ &= e(g,g)^{\alpha s} e(g,h)^{(a+id+b\cdot r)ts}. \end{aligned}$$

Then M is recovered by $C \cdot D/E$.

The IND-CPA security of the construction above can be proved similarly as the main construction in Section IV, and the traceability can be proved by the proof ideas of the main construction in Section IV and the signature scheme in [18].

VI. Conclusion and Future Work

In this work, we constructed a Traceable CP-ABE system which achieved the same degree of high expressiveness (i.e., support ciphertext policies in any form of monotone access structures), efficiency and security level as one of the best existing (non-traceable) CP-ABE systems. In particular, in our proposed scheme, given a decryption key, the tracing algorithm is able to find out the original key owner, and therefore, can deter a malicious key owner to leak his decryption key for whatever motivation he has (e.g., for financial gain) without getting caught. This system is the first traceable CP-ABE system that supports any monotone access structures and achieving adaptive security in the standard model. The cost of achieving traceability in our system is also very low (i.e., two additional elements in a decryption key and one additional element in each ciphertext).

Black-Box Traceable CP-ABE Supporting Any Monotone Access Structures: Our proposed system is the first Traceable CP-ABE system supporting any monotone access structures. However, traceability in our system is in the white-box model (i.e., the cheating user is selling his/her decryption key or masqueraded/modified decryption key to some buyers, and the buyers will use the existing decryption equipment or module to perform decryption), in other words, the key to be traced is assumed to be well-formed and can be used by a well-formed decryption algorithm as input to decrypt ciphertexts. Consequently, our proposed scheme is not secure with respect to black-box traceability: one can tweak the decryption algorithm and integrate it with some randomized keys to build a decryption equipment while selling the whole thing as a black-box device to a buyer. We believe that due to the practical needs, an interesting open problem is to construct a Traceable CP-ABE system supporting any monotone access structures with traceability in the black-box model. From the views of both practice and theory, flexible expressiveness is one of the most important features of CP-ABE, and our work in this paper shows an encouraging result towards the construction of a Traceable CP-ABE with traceability in the black-box model

[5] In the signature scheme, the signature for a message $m \in \mathbb{Z}_p^*$ is $(r, g^{1/(a+b\cdot r+m)}) \in \mathbb{Z}_p^* \times G$, where (a,b) is the secret key for signing and (g, g^a, g^b) is the public key for verification.

and the support of flexible expressiveness (i.e., any monotone access structures) simultaneously. Although black-box traitor tracing systems in broadcast encryption has been proposed, but it seems that the essential difference between the settings of CP-ABE and broadcast encryption is preventing the techniques in broadcast encryption from being applied to CP-ABE. To obtain a CP-ABE system which is expressive and black-box traceable simultaneously, new techniques are expected, and we leave it as our future work.

Appendix A
An Analysis of the Tracing Algorithms in [14], [15]

Each user in [15] is assigned an identity $\mathsf{ID} \in \{0,1\}^l$, and each bit of ID is considered as an attribute. As the ciphertext-policy in the system is a single **AND** gate with wildcard, for the policy of *normal ciphertexts* all the identity-bit-attribute values are set to "*" indicating "do not care", and for the policy of *tracing ciphertexts* the identity-bit-attribute values can be set to "0", "1" or "*" so that the tracing algorithm can pinpoint the bits of the ID in the decryption-box bit-by-bit.

However, as the attacker is allowed to query multiple decryption keys, a decryption-box can be built as follows: The decryption-box contains two keys of identities ID_1 and ID_2, respectively. To decrypt a ciphertext, the decryption-box will compare the plaintexts obtained from decryption using the two keys, and only if they are the same, the decryption-box outputs the plaintext, otherwise it outputs a random message.

We can see that using a decryption-box based on the above "compare-before-output" technique can avoid the tracing algorithm in [15] from identifying the malicious user. The tracing algorithm can find out the bits on the positions where the two identities have the same value, but can obtain nothing on the positions with different bit values. Assuming there are w "1" bits in $\mathsf{ID}_1 \oplus \mathsf{ID}_2$, the tracing algorithm in [15] will trace to a space of size 2^w rather than a valid identity. In the worst case of $\mathsf{ID}_1 \oplus \mathsf{ID}_2 = 111\ldots111$, the tracing algorithm obtains nothing.

In [14] each user is assigned an identity. The bits of the identity are also considered as attributes and are embedded in the user's key policy by an **AND** gate. The attributes of the ciphertext contain the target receiver's identity-bit-attribute values, which are set to "***...***" indicating "do not care" for *normal ciphertexts* and are set to an exact identity for *tracing ciphertexts*. In [14] there is a restriction that no pair of users have exactly the same access privilege, but it is noted that a new key of policy "(A **AND** B)" for user ID_1 can be created from a key of policy "(A **AND** B) **OR** C" for user ID_1. Similarly, a new key of policy "(A **AND** B)" for user ID_2 can be created from a key of policy "(A **AND** B) **OR** D" for user ID_2. Thus, the above "compare-before-output" decryption-box can also avoid the tracing algorithm in [14] from identifying the malicious user, even under the restriction above.

References

[1] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. EUROCRYPT*, R. Cramer, Ed., 2005, vol. 3494, pp. 457–473, ser. Lecture Notes in Computer Science, Springer.

[2] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. CRYPTO*, 1984, pp. 47–53.

[3] D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing," in *Proc. CRYPTO*, J. Kilian, Ed., 2001, vol. 2139, pp. 213–229, ser. Lecture Notes in Computer Science, Springer.

[4] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. ACM Conf. Computer and Communications Security*, A. Juels, R. N. Wright, and S. D. C. di Vimercati, Eds., 2006, pp. 89–98, ACM.

[5] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proc. ACM Conf. Computer and Communications Security*, P. Ning, S. D. C. di Vimercati, and P. F. Syverson, Eds., 2007, pp. 195–203, ACM.

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

[7] L. Cheung and C. C. Newport, "Provably secure ciphertext policy abe," in *Proc. ACM Conf. Computer and Communications Security*, P. Ning, S. D. C. di Vimercati, and P. F. Syverson, Eds., 2007, pp. 456–465, ACM.

[8] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in *Proc. ICALP (2)*, L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, Eds., 2008, vol. 5126, pp. 579–591, ser. Lecture Notes in Computer Science, Springer.

[9] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Proc. Public Key Cryptography*, D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, Eds., 2011, vol. 6571, pp. 53–70, ser. Lecture Notes in Computer Science, Springer.

[10] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Proc. EUROCRYPT*, H. Gilbert, Ed., 2010, vol. 6110, pp. 62–91, ser. Lecture Notes in Computer Science, Springer.

[11] T. Okamoto and K. Takashima, "Fully secure functional encryption with general relations from the decisional linear assumption," in *Proc. CRYPTO*, T. Rabin, Ed., 2010, vol. 6223, pp. 191–208, ser. Lecture Notes in Computer Science, Springer.

[12] M. J. Hinek, S. Jiang, R. Safavi-Naini, and S. F. Shahandashti, Attribute-Based Encryption With Key Cloning Protection Cryptology ePrint Archive, Report 2008/478, 2008 [Online]. Available: http://eprint.iacr.org/

[13] J. Li, K. Ren, and K. Kim, A2be: Accountable Attribute-Based Encryption for Abuse Free Access Control Cryptology ePrint Archive, Report 2009/118, 2009 [Online]. Available: http://eprint.iacr.org/

[14] S. Yu, K. Ren, W. Lou, and J. Li, Social Informatics and Telecommunications Engineering, "Defending against key abuse attacks in kp-abe enabled broadcast systems," in *Proc. SecureComm*, Y. Chen, T. Dimitriou, and J. Zhou, Eds., 2009, vol. 19, pp. 311–329, ser. Lecture Notes of the Institute for Computer Sciences, Springer.

[15] 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. ASIACCS*, B. S. N. Cheung, L. C. K. Hui, R. S. Sandhu, and D. S. Wong, Eds., 2011, pp. 386–390, ACM.

[16] V. Goyal, "Reducing trust in the pkg in identity based cryptosystems," in *Proc. CRYPTO*, A. Menezes, Ed., 2007, vol. 4622, pp. 430–447, ser. Lecture Notes in Computer Science, Springer.

[17] V. Goyal, S. Lu, A. Sahai, and B. Waters, "Black-box accountable authority identity-based encryption," in *Proc. ACM Conf. Computer and Communications Security*, P. Ning, P. F. Syverson, and S. Jha, Eds., 2008, pp. 427–436, ACM.

[18] D. Boneh and X. Boyen, "Short signatures without random oracles," in *Proc. EUROCRYPT*, C. Cachin and J. Camenisch, Eds., 2004, vol. 3027, pp. 56–73, ser. Lecture Notes in Computer Science, Springer.

[19] M. Chase, "Multi-authority attribute based encryption," in *Proc. TCC*, S. P. Vadhan, Ed., 2007, vol. 4392, pp. 515–534, ser. Lecture Notes in Computer Science, Springer.

[20] H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure threshold multi authority attribute based encryption without a central authority," in *Proc. INDOCRYPT*, D. R. Chowdhury, V. Rijmen, and A. Das, Eds., 2008, vol. 5365, pp. 426–436, ser. Lecture Notes in Computer Science, Springer.

[21] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proc. ACM Conf. Computer and Communications Security*, E. Al-Shaer, S. Jha, and A. D. Keromytis, Eds., 2009, pp. 121–130, ACM.

[22] A. B. Lewko and B. Waters, “Decentralizing attribute-based encryption,” in *Proc. EUROCRYPT*, K. G. Paterson, Ed., 2011, vol. 6632, pp. 568–588, ser. Lecture Notes in Computer Science, Springer.

[23] Z. Liu, Z. Cao, Q. Huang, D. S. Wong, and T. H. Yuen, “Fully secure multi-authority ciphertext-policy attribute-based encryption without random oracles,” in *Proc. ESORICS*, V. Atluri and C. Díaz, Eds., 2011, vol. 6879, pp. 278–297, ser. Lecture Notes in Computer Science, Springer.

[24] B. Chor, A. Fiat, and M. Naor, “Tracing traitors,” in *Proc. CRYPTO*, Y. Desmedt, Ed., 1994, vol. 839, pp. 257–270, ser. Lecture Notes in Computer Science, Springer.

[25] D. Boneh, A. Sahai, and B. Waters, “Fully collusion resistant traitor tracing with short ciphertexts and private keys,” in *Proc. EUROCRYPT*, S. Vaudenay, Ed., 2006, vol. 4004, pp. 573–592, ser. Lecture Notes in Computer Science, Springer.

[26] A. Beimel, “Secure Schemes for Secret Sharing and Key Distribution,” Ph.D. Dissertation, Israel Institute of Technology, Technion, Haifa, Israel, 1996.

[27] D. Boneh and X. Boyen, “Efficient selective-id secure identity-based encryption without random oracles,” in *Proc. EUROCRYPT*, C. Cachin and J. Camenisch, Eds., 2004, vol. 3027, pp. 223–238, ser. Lecture Notes in Computer Science, Springer.

[28] D. Boneh, E.-J. Goh, and K. Nissim, “Evaluating 2-DNF formulas on ciphertexts,” in *TCC*, J. Kilian, Ed., 2005, vol. 3378, pp. 325–341, ser. Lecture Notes in Computer Science, Springer.

[29] A. B. Lewko and B. Waters, “New techniques for dual system encryption and fully secure hibe with short ciphertexts,” in *Proc. TCC*, D. Micciancio, Ed., 2010, vol. 5978, pp. 455–479, ser. Lecture Notes in Computer Science, Springer.

[30] R. Canetti, S. Halevi, and J. Katz, “Chosen-ciphertext security from identity-based encryption,” in *Proc. EUROCRYPT*, C. Cachin and J. Camenisch, Eds., 2004, vol. 3027, pp. 207–222, ser. Lecture Notes in Computer Science, Springer.

Zhen Liu received the B.S. and M.S. degrees from Shanghai Jiao Tong University, in 1999 and 2002, respectively. He is currently a Ph.D. candidate for the Ph.D. program jointly offered by City University of Hong Kong and Shanghai Jiao Tong University.

His research interests include applied cryptography and information security, in particular, encryption and signature schemes.

Zhenfu Cao (A’09–M’09–SM’10) received the B.Sc. degree in computer science and technology and the Ph.D. degree in mathematics from Harbin Institute of Technology, Harbin, China, in 1983 and 1999, respectively.

He was exceptionally promoted to Associate Professor in 1987 and became a Professor in 1991. He is currently a Distinguished Professor and the Director of the Trusted Digital Technology Laboratory, Shanghai Jiao Tong University, Shanghai, China. He also serves as a member of the expert panel of the National Nature Science Fund of China.

Prof. Cao is actively involved in the academic community, serving as Committee/Session Chair and program committee member for several international conference committees, as follows: the IEEE Global Communications Conference (since 2008), the IEEE International Conference on Communications (since 2008), the International Conference on Communications and Networking in China (since 2007), etc. He is the Associate Editor of *Computers and Security* (Elsevier), an Editorial Board member of *Fundamenta Informaticae* (IOS) and *Peer-to-Peer Networking and Applications* (Springer-Verlag), and Guest Editor of the *Special Issue on Wireless Network Security, Wireless Communications and Mobile Computing* (Wiley), 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, and a corecipient of the 2007 IEEE International Conference on Communications (Computer and Communications Security Symposium) Best Paper Award in 2007. He also received seven awards granted by the National Ministry and governments of provinces such as the first prize of the Natural Science Award from the Ministry of Education.

Duncan S. Wong received the B.Eng. degree from the University of Hong Kong in 1994, the M.Phil. degree from the Chinese University of Hong Kong in 1998, and the Ph.D. degree from Northeastern University, Boston, MA, in 2002.

He is currently an associate professor in the Department of Computer Science at the City University of Hong Kong. His primary research interest is cryptography; in particular, cryptographic protocols, encryption and signature schemes, and anonymous systems. He is also interested in other topics in information security, such as network security, wireless security, database security, and security in cloud computing.