Traceable CP-ABE: How to Trace Decryption Devices Found in the Wild

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

***Abstract*— In Ciphertext-policy attribute-based encryption (CP-ABE), ciphertexts are associated with access policies, which do not have to contain the identities of eligible receivers, and attributes are shared by multiple users. CP-ABE is useful for providing fine-grained access control on encrypted data. However, it also has a practicality concern that a malicious user, with his attributes shared with other users, might leak his decryption privilege as a decryption blackbox, for some financial gain or other incentives, as there is little risk of getting caught. There are two types of decryption blackboxes that reflect different practical scenarios. A key-like decryption blackbox is associated with an attribute set $S_{\mathcal{D}}$ and can decrypt ciphertexts with access policies satisfied by $S_{\mathcal{D}}$. A policy-specific decryption blackbox is associated with an access policy $\mathbb{A}_{\mathcal{D}}$ and can decrypt ciphertexts with $\mathbb{A}_{\mathcal{D}}$. Policy-specific decryption blackbox has weaker decryption capacity than key-like decryption blackbox, but tracing it is deemed to be more difficult. In the preliminary version (in CCS 2013) of this paper, we proposed a new CP-ABE scheme which is adaptively traceable against key-like decryption blackbox. The scheme has sublinear overhead, which is the most efficient one to date supporting fully collusion-resistant blackbox traceability. The scheme is fully secure in the standard model, and supports any monotonic access structures. In this paper, we further show that the scheme is also selectively traceable against policy-specific decryption blackbox. Furthermore, and more importantly, we prove a general statement that if a CP-ABE scheme is (selectively) traceable against policy-specific decryption blackbox, it is also (selectively) traceable against key-like decryption blackbox, which implies that we now only need to focus on building CP-ABE schemes which are traceable against policy-specific decryption blackbox.**

***Index Terms*— Traceability, CP-ABE, blackbox, key-like, policy-specific.**

I. INTRODUCTION

CIPHERTEXT-POLICY Attribute-Based Encryption (CP-ABE), introduced by Goyal et al. [1], is a versatile one-to-many encryption primitive which enables fine-grained access control over encrypted data. Suppose Alice wants to encrypt a message for all PhD students and alumni in the Department of Mathematics, but she does not know or is not able to find out the identities of all the eligible receivers, and the set of eligible receivers may also be dynamic. Intuitively, Alice, in this example, is to encrypt a message under "(Mathematics AND (PhD Student OR Alumni))", which is an *access policy* defined over descriptive *attributes*, so that only those receivers who have their decryption keys associated with the attributes which satisfy this policy can decrypt.

Traditional public key encryption and identity-based encryption [2], [3] are inefficient to realize the requirement in the example above as they are for one-to-one encryption. Broadcast Encryption (BE) [4] may not be suitable either as the encryptor in BE has to know and specify the identities/indices of the receivers in advance. In CP-ABE, an authority issues different decryption keys to each user based on the attributes the user possesses. During encryption, an encryptor specifies an access policy for the resulting ciphertext. If and only if a receiver's attributes satisfy the access policy of the ciphertext can the receiver decrypt the ciphertext.

Among the CP-ABE schemes recently proposed [5]–[13], progress has been made on the schemes' security, access policy expressivity, and efficiency. In [12], Lewko and Waters proposed a new proofing technique and obtained a CP-ABE which is fully secure (i.e. provably secure against adaptive adversaries in the standard model), highly expressive (i.e. supporting any monotonic access structures) and efficient, and additionally eliminates the one-use restriction that previous schemes [9], [10] have. Specifically, the security proof in [9] and [10] relies on the *one-use restriction* that a single attribute can only be used once in a policy, and directly extending the schemes in [9] and [10] to allow attribute reuse would incur significant tradeoffs in efficiency.

One of the major practicality issues of CP-ABE to date is the lack of effective solutions to identify malicious users which intentionally expose their secret decryption keys, for example, for financial gain. Due to the nature of CP-ABE, access policies associated with the ciphertexts do not have to contain the exact identities of eligible receivers. Instead, access policies are role-based and the attributes are generally *shared* by multiple users. For example, both Bob (with attributes {Bob, Alumni, Mathematics}) and Tom (with attributes {Tom, Alumni, Mathematics}) could share a decryption key corresponding to attributes {Alumni, Mathematics} and be able to decrypt the ciphertext in the example above,

Manuscript received April 22, 2014; revised August 4, 2014; accepted October 6, 2014. Date of publication October 16, 2014; date of current version December 5, 2014. This work was supported in part by the National Natural Science Foundation of China under Grant 61161140320, Grant 61371083, and Grant 61373154, in part by the Prioritized Development Projects, Specialized Research Fund for the Doctoral Program of Higher Education of China, under Grant 20130073130004, and in part by the Research Grants Council, Hong Kong, under Project CityU 123913. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Liqun Chen. *(Corresponding author: Zhenfu Cao.)*

Z. Liu and D. S. Wong are with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: zhenliu7@cityu.edu.hk; duncan@cityu.edu.hk).

Z. Cao is with the Shanghai Key Lab for Trustworthy Computing, East China Normal University, Shanghai 200062, China (e-mail: zfcao@sei.ecnu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2014.2363562

while the key has no identity information. As a result, a malicious user, with his attributes shared with multiple other users, might be tempted to leak a decryption key or some *decryption privilege* in the form of a decryption blackbox/device in which the decryption key is embedded, for some incentives, as there is little risk of getting caught. This is an interesting and realistic problem as leaking a decryption key or a more advanced decryption blackbox may entail financial gain and even better, the malicious user has very little risk of getting caught.

To address this problem, we require a CP-ABE system to support *traceability*. There are two levels of traceability. Level one is *Whitebox Traceability*, by which given a well-formed decryption key, a *tracing algorithm* can find out the user which owns the key. This also includes a scenario that a malicious user sells a well-formed decryption key which is not identical but is created from the user's key. A whitebox traceable CP-ABE scheme, with the same security, access policy expressivity and efficiency as the CP-ABE in [9], has been proposed in [14]. However, a whitebox traceable CP-ABE scheme may not be able to identify a malicious user which makes up a decryption blackbox.

Level two is *Blackbox Traceability*, by which given a *decryption blackbox/device*, while the decryption key and even the decryption algorithm could be hidden in the blackbox, the tracing algorithm, which treats the decryption blackbox as an oracle, can still find out the malicious user whose key has been used in constructing the decryption blackbox. Below are two types of decryption blackboxes that reflect different practical scenarios [15].

Key-Like Decryption Blackbox for Sale. Using his decryption key (or the decryption keys from multiple colluding malicious users), a malicious user builds a decryption blackbox (i.e. a CP-ABE Decryption Blackbox) and sells it on eBay for financial gain. To invalidate the possible whitebox tracing algorithms, the seller keeps the embedded decryption keys and (possibly complicated) algorithms hidden and the device works as a decryption blackbox. Then, to attract potential buyers, the seller describes and advertises that the decryption blackbox functions like a decryption key associated with an attribute set $S_{\mathcal{D}}$, i.e., if a ciphertext access policy can be satisfied by $S_{\mathcal{D}}$, the ciphertext can be decrypted by the blackbox. We call such a decryption blackbox a ***key-like decryption blackbox***, which could be deemed to be quite attractive to potential buyers, and the resulting financial gain could be a big incentive for malicious users to build and sell such a blackbox.

Policy-Specific Decryption Blackbox ("Found-in-the-Wild"). A law enforcement agency gets a warrant to search a suspect's computer and finds a decryption blackbox. As the suspect might try to destroy evidence, the explicit description of the blackbox's (decryption) ability might be gone, while the law enforcement agency only has certain clue on the certain access policy associated to the ciphertexts that the blackbox can decrypt. We refer to such a decryption blackbox as a ***policy-specific decryption blackbox***. Unlike a key-like decryption blackbox, which has an attribute set $S_{\mathcal{D}}$ associated as the alleged decryption capability being advertised, a policy-specific decryption blackbox is only known to be able to decrypt ciphertexts with some specific policy $\mathbb{A}_{\mathcal{D}}$.

A. Our Results

In the preliminary version [15] of this paper, we proposed a new CP-ABE which is fully secure (against adaptive adversaries in the standard model), highly expressive (i.e. supporting any monotonic access structures), and not one-use restricted. Furthermore, the new scheme is fully collusion-resistant blackbox traceable against key-like decryption blackbox, namely, among all the colluding users who pull together their decryption keys to build the decryption blackbox, the tracing algorithm of this scheme can find out at least one of them. In addition, the traceability is public, that is, anyone can run the tracing algorithm without any additional secret input.

When compared with the most efficient conventional (non-traceable) highly expressive CP-ABE currently available (for example [12]), this new scheme *adds* the public and fully collusion-resistant blackbox traceability with the price of adding only $O(\sqrt{\mathcal{K}})$ elements in the ciphertext and public key, rather than expanding the sizes linearly with $\mathcal{K}$, which is the number of users in the system.

In this paper, which serves as the extended version of [15], we move forward to investigate the policy-specific decryption blackbox. New contributions in this paper include:

1) We prove that the CP-ABE scheme proposed in the preliminary version [15] is selectively traceable against policy-specific decryption blackbox. Combining the results in our preliminary version [15] and this new result, we now have a fully secure and highly expressive CP-ABE scheme which is adaptively traceable against key-like decryption blackbox (as proved in [15]) and selectively traceable against policy-specific decryption blackbox (as proved in this paper).
2) We prove a general statement that a CP-ABE scheme which is (selectively) traceable against policy-specific decryption blackbox is also (selectively) traceable against key-like decryption blackbox. This implies that now we only need to focus on constructing secure CP-ABE schemes which are traceable against policy-specific decryption blackbox.

In Sec. II, we first review the definitions of CP-ABE and its traceability against key-like decryption blackbox, which were given in [15], then we define traceability against policy-specific decryption blackbox and show that a secure CP-ABE with (selective) traceability against policy-specific decryption blackbox is also (selectively) traceable against key-like decryption blackbox.

In Sec. III, we review the definitions of Augmented CP-ABE (AugCP-ABE) and its ***message-hiding*** property, and redefine its ***index-hiding*** property while referring to the index-hiding property defined in [15] as ***weak index-hiding***, which is limited to AND policy, while the new definition given in this paper is for arbitrary access policies.

Furthermore, we show that an AugCP-ABE with message-hiding and (selective) index-hiding properties can be

TABLE I

EFFICIENCY AND TRACEABILITY COMPARISON

	Public Key Size	Ciphertext Size	Private Key Size	Pairing Computation in Decryption	Traceability against Key-like Blackbox	Traceability against Policy-specific Blackbox
[12]	$\lvert\mathcal{U}\rvert + 4$	$2l + 3$	$\lvert S\rvert + 3$	$2\lvert I\rvert + 2$	×	×
[14]	$\lvert\mathcal{U}\rvert + 4$	$2l + 3$	$\lvert S\rvert + 4$	$2\lvert I\rvert + 1$	×	×
[15] and this work	$\lvert\mathcal{U}\rvert + 3 + 4\sqrt{\mathcal{K}}$	$2l + 17\sqrt{\mathcal{K}}$	$\lvert S\rvert + 4$	$2\lvert I\rvert + 10$	adaptive, public fully collusion-resistant	selective, public fully collusion-resistant

[1] The CP-ABE construction of this paper is the same as that in [15]. All the three CP-ABE schemes are fully secure and highly expressive (i.e. supporting any monotonic access structures).

[2] For the CP-ABE scheme in [15] (and this work), the adaptive traceability against key-like decryption blackbox is proved in [15], whereas the selective traceability against policy-specific decryption blackbox is proved in this paper.

[3] Let $|\mathcal{U}|$ be the size of the attribute universe, l the size of an access policy, $|S|$ the size of the attribute set of a private key, and $|I|$ the number of attributes in a decryption key that satisfies a ciphertext's access policy.

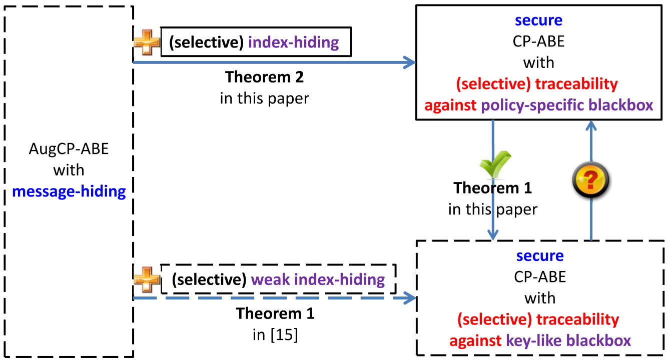


Fig. 1. The relations between AugCP-ABE and various Traceable CP-ABE types

transformed into a secure CP-ABE with (selective) traceability against policy-specific decryption blackbox. Note that a related transformation given in [15] shows that an AugCP-ABE with message-hiding and (selective) weak index-hiding implies a secure CP-ABE with (selective) traceability against key-like decryption blackbox. Informally, these two transformations are evidences that traceability against policy-specific decryption blackbox is more difficult to achieve when compared with traceability against key-like decryption blackbox, as achieving the former one requires an AugCP-ABE with the index-hiding property (new definition given in this paper) while the latter one only needs the weak index-hiding property. The relations among AugCP-ABE, CP-ABE with traceability against key-like decryption blackbox, and CP-ABE with traceability against policy-specific decryption blackbox are illustrated in Fig 1, where the dashed parts were given in [15] and the other parts are new in this paper.

In Sec. IV, we revisit the AugCP-ABE construction in [15], which has been proven to satisfy weak index-hiding, and show that the construction is also selectively index-hiding. In [15], we already showed that the derived CP-ABE from the AugCP-ABE construction is traceable against key-like decryption blackbox. In this paper, we further show that the derived CP-ABE is also selectively traceable against policy-specific decryption blackbox. Table I compares our scheme in [15] (and this work) with that in [12] and [14] as all of these schemes are fully secure and highly expressive.

II. CP-ABE WITH TRACEABILITY

We review the definition of CP-ABE which is based on the conventional (non-traceable) CP-ABE (e.g. [9], [12]) with the exception that in our 'functional' definition, we explicitly assign and identify users using unique indices. Then we define traceability against key-like decryption blackbox and policy-specific decryption blackbox. At last, we show that a secure CP-ABE with traceability against policy-specific decryption blackbox implies traceability against key-like decryption blackbox.

A. CP-ABE

A Ciphertext-Policy Attribute-Based Encryption (CP-ABE) system consists of four algorithms:

- *Setup* $(\lambda, \mathcal{U}, \mathcal{K}) \rightarrow (\mathsf{PP}, \mathsf{MSK})$*:* The algorithm takes as input a security parameter λ, the attribute universe $\mathcal{U}$, and the number of users $\mathcal{K}$ in the system, then runs in polynomial time in λ, and outputs the public parameter PP and a master secret key MSK.
- *KeyGen* $(\mathsf{PP}, \mathsf{MSK}, S) \rightarrow \mathsf{SK}_{k,S}$*:* The algorithm takes as input the public parameter PP, the master secret key MSK, and an attribute set S, and outputs a private decryption key $\mathsf{SK}_{k,S}$, which is assigned and identified by a unique index $k \in \{1, \ldots, \mathcal{K}\}$.
- *Encrypt* $(\mathsf{PP}, M, \mathbb{A}) \rightarrow CT$*:* The algorithm takes as input the public parameter PP, a message M, and an access policy $\mathbb{A}$ over $\mathcal{U}$, and outputs a ciphertext CT such that only users whose attributes satisfy $\mathbb{A}$ can recover M. $\mathbb{A}$ is implicitly included in CT.
- *Decrypt* $(\mathsf{PP}, CT, \mathsf{SK}_{k,S}) \rightarrow M$ *or* $\perp$*:* The algorithm takes as input the public parameter PP, a ciphertext CT, and a private key $\mathsf{SK}_{k,S}$. If S satisfies CT's access policy, the algorithm outputs a message M, otherwise it outputs $\perp$ indicating the failure of decryption.

The security of a CP-ABE system is defined using the following message-hiding game.

$\mathsf{Game}_{\mathsf{MH}}$. The Message-hiding game is defined between a challenger and an adversary $\mathcal{A}$ as follows:

- *Setup:* The challenger runs $\mathsf{Setup}(\lambda, \mathcal{U}, \mathcal{K})$ and gives the public parameter PP to $\mathcal{A}$.
- *Phase 1:* For $i = 1$ to q_1, $\mathcal{A}$ adaptively submits (k_i, S_{k_i}), and the challenger responds with $\mathsf{SK}_{k_i, S_{k_i}}$.

- *Challenge:* $\mathcal{A}$ submits two equal-length messages M_0, M_1 and an access policy $\mathbb{A}^*$. The challenger flips a random coin $b \in \{0, 1\}$, and gives $\mathcal{A}$ an encryption of M_b under $\mathbb{A}^*$.
- *Phase 2:* For $i = q_1 + 1$ to q, $\mathcal{A}$ adaptively submits (k_i, S_{k_i}), and the challenger responds with $\mathsf{SK}_{k_i, S_{k_i}}$.
- *Guess:* $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ for b.

$\mathcal{A}$ wins the game if $b' = b$ under the **restriction** that $\mathbb{A}^*$ cannot be satisfied by any of the queried attribute sets $S_{k_1}, \ldots, S_{k_q}$. The advantage of $\mathcal{A}$ is defined as $\mathsf{MHAdv}_{\mathcal{A}} = |\Pr[b' = b] - \frac{1}{2}|$.

Definition 1: A $\mathcal{K}$-user CP-ABE system is secure if for all polynomial-time adversaries $\mathcal{A}$ the advantage $\mathsf{MHAdv}_{\mathcal{A}}$ *is negligible in λ.*

It is worth noticing that: (1) although the index of each user is assigned by the KeyGen algorithm, to capture the security that an attacker can adaptively choose keys to corrupt, we allow the adversary to specify the index when he makes a key query, i.e., for $i = 1$ to q, the adversary submits (index, attribute set) pair (k_i, S_{k_i}) to query a private key for attribute set S_{k_i}, where $q \leq \mathcal{K}$, $k_i \in \{1, \ldots, \mathcal{K}\}$, and $k_i \neq k_j \; \forall 1 \leq i \neq j \leq q$ (this is to guarantee that each user/key can be *uniquely* identified by an index); and (2) for $k_i \neq k_j$ we do not require $S_{k_i} \neq S_{k_j}$, i.e., different users/keys may have the same attribute set. We remark that these two points apply to the rest of the paper.

B. Traceability Against Key-Like Decryption Blackbox

A key-like decryption blackbox $\mathcal{D}$ in the setting of CP-ABE functions like a decryption key as the name implies. The blackbox is associated with an attribute set $S_{\mathcal{D}}$ and can decrypt ciphertxets with access policies satisfied by $S_{\mathcal{D}}$. Also, a key-like decryption blackbox is viewed as a probabilistic circuit, and it does not need to be perfect, namely, we only require it to be able to decrypt with non-negligible success probability. Key-like decryption blackbox reflects the following practical scenario.

Key-Like Decryption Blackbox for Sale. *An adversary (i.e. seller) describes a key-like decryption blackbox $\mathcal{D}$ with a non-empty attribute set $S_{\mathcal{D}}$ and a non-negligible probability value ϵ (i.e. $0 < \epsilon \leq 1$ is polynomially related to λ), and advertises that for any access policy $\mathbb{A}$, if it can be satisfied by $S_{\mathcal{D}}$, this blackbox $\mathcal{D}$ can decrypt the corresponding ciphertext associated with $\mathbb{A}$ with probability at least ϵ.* Note that ϵ is the lower-bound of $\mathcal{D}$'s decryption ability, e.g., suppose $\mathbb{A}_1$ is a ciphertext's access policy satisfied by $S_{\mathcal{D}}$ and $\mathcal{D}$ can decrypt the ciphertext with probability 0.1, even if $\mathcal{D}$ can decrypt ciphertexts under other access policies (satisfied by $\mathcal{D}$) with probability 1, the seller can only declare an $\epsilon \leq 0.1$. Obviously for some attribute set $S_{\mathcal{D}}$, ϵ is closer to 1, which implies that the decryption ability of $\mathcal{D}$ is closer to that of a private key with attribute set $S_{\mathcal{D}}$, and hence $\mathcal{D}$ is more attractive to potential buyers.

We now define the tracing algorithm and traceability against key-like decryption blackbox.

$\mathsf{Trace}_{\mathsf{KL}}^{\mathcal{D}}(\mathsf{PP}, S_{\mathcal{D}}, \epsilon) \rightarrow \mathbb{K}_T \subseteq \{1, \ldots, \mathcal{K}\}$. $\mathsf{Trace}_{\mathsf{KL}}$ *is an oracle algorithm that interacts with a key-like decryption blackbox $\mathcal{D}$. By given the public parameter* PP, *a non-empty attribute set $S_{\mathcal{D}}$, and a probability value (lower-bound) ϵ, the algorithm runs in time polynomial in λ and $1/\epsilon$, and outputs an index set $\mathbb{K}_T \subseteq \{1, \ldots, \mathcal{K}\}$ which identifies the set of malicious users. Note that ϵ has to be polynomially related to λ.*

The following $\mathsf{Tracing\ Game}$ captures the notion of **fully collusion-resistant traceability**, where the adversary has access to an arbitrary number of private keys (i.e. colluding users).

$\mathsf{Game}_{\mathsf{TR}}^{\mathsf{KL}}$. The $\mathsf{Tracing\ Game}$ is defined between a challenger and an adversary $\mathcal{A}$ as follows:

- *Setup:* The challenger runs $\mathsf{Setup}(\lambda, \mathcal{U}, \mathcal{K})$ and gives the public parameter PP to $\mathcal{A}$.
- *Key Query:* For $i = 1$ to q, $\mathcal{A}$ adaptively submits (k_i, S_{k_i}), and gets $\mathsf{SK}_{k_i, S_{k_i}}$ from the challenger.
- *(Key-Like) Decryption Blackbox Generation:* $\mathcal{A}$ outputs a decryption blackbox $\mathcal{D}$ associated with a non-empty attribute set $S_{\mathcal{D}} \subseteq \mathcal{U}$ and a non-negligible probability (lower-bound) value ϵ.
- *Tracing:* The challenger runs $\mathsf{Trace}_{\mathsf{KL}}^{\mathcal{D}}(\mathsf{PP}, S_{\mathcal{D}}, \epsilon)$ to obtain an index set $\mathbb{K}_T \subseteq \{1, \ldots, \mathcal{K}\}$.

Let $\mathbb{K}_{\mathcal{D}} = \{k_i | 1 \leq i \leq q\}$ be the index set of keys corrupted by the adversary. We say that the adversary $\mathcal{A}$ wins the game if the following conditions hold:

1) For any access policy $\mathbb{A}$ that is satisfied by $S_{\mathcal{D}}$, we have

$$\Pr[\mathcal{D}(\mathsf{Encrypt}(\mathsf{PP}, M, \mathbb{A})) = M] \geq \epsilon,$$

where the probability is taken over the random choices of message M and the random coins of $\mathcal{D}$. A decryption blackbox satisfying this condition is said to be a *useful key-like decryption blackbox.*

2) $\mathbb{K}_T = \emptyset$, or $\mathbb{K}_T \not\subseteq \mathbb{K}_{\mathcal{D}}$, or $(S_{\mathcal{D}} \not\subseteq S_{k_t} \; \forall k_t \in \mathbb{K}_T)$.

We denote by $\mathsf{TRAdv}_{\mathcal{A}}^{\mathsf{KL}}$ the probability that adversary $\mathcal{A}$ wins this game.

Definition 2: A $\mathcal{K}$-user CP-ABE system is traceable against key-like decryption blackbox if for all polynomial-time adversaries $\mathcal{A}$ the advantage $\mathsf{TRAdv}_{\mathcal{A}}^{\mathsf{KL}}$ *is negligible in λ.*

We say that a $\mathcal{K}$-user CP-ABE system is *selectively* traceable against key-like decryption blackbox if we add an **Init** stage before **Setup** for the adversary to commit the attribute set $S_{\mathcal{D}}$.

Remark: For a useful key-like decryption blackbox $\mathcal{D}$, the traced $\mathbb{K}_T$ must satisfy $(\mathbb{K}_T \neq \emptyset) \wedge (\mathbb{K}_T \subseteq \mathbb{K}_{\mathcal{D}}) \wedge (\exists k_t \in \mathbb{K}_T \; s.t. \; S_{k_t} \supseteq S_{\mathcal{D}})$. (1) $(\mathbb{K}_T \neq \emptyset) \wedge (\mathbb{K}_T \subseteq \mathbb{K}_{\mathcal{D}})$ captures the preliminary traceability that the $\mathsf{Trace}_{\mathsf{KL}}$ algorithm can extract at least one malicious user and the coalition of malicious users cannot frame any innocent user. (2) $(\exists k_t \in \mathbb{K}_T \; s.t. \; S_{k_t} \supseteq S_{\mathcal{D}})$ captures the *strong traceability* that the $\mathsf{Trace}_{\mathsf{KL}}$ algorithm can extract at least one malicious user whose private key enables $\mathcal{D}$ to have the decryption ability corresponding to $S_{\mathcal{D}}$, i.e., whose attribute set is a superset of $S_{\mathcal{D}}$.

C. Traceability Against Policy-Specific Decryption Blackbox

A policy-specific decryption blackbox $\mathcal{D}$ in the setting of CP-ABE is viewed as a probabilistic circuit that can decrypt ciphertexts generated under some specific access policy.

In particular, a policy-specific decryption blackbox $\mathcal{D}$ is described with an access policy $\mathbb{A}_{\mathcal{D}}$ and a non-negligible probability value ϵ (i.e. $0 < \epsilon \leq 1$ is polynomially related to λ), and this blackbox $\mathcal{D}$ can decrypt the ciphertexts generated under $\mathbb{A}_{\mathcal{D}}$ with probability at least ϵ. Policy-specific decryption blackbox has weaker decryption ability than key-like decryption blackbox, as it can decrypt only ciphertexts with a specific access policy. This type of blackboxes reflects a traitor's attempt of making the tracing efforts of the law enforcement in vain.[1] In addition, policy-specific decryption blackbox also reflects the following practical scenario.

Decryption Blackbox Found-in-the-Wild. *A law enforcement agency gets a warrant to search a suspect's computer and finds a decryption blackbox. As the suspect might try to destroy evidence, the explicit description of the blackbox's (decryption) ability might be gone, while the law enforcement agency only has certain clue on the certain access policy associated to the ciphertexts that the blackbox can decrypt.*

We now define the tracing algorithm and traceability against policy-specific decryption blackbox.

$\mathsf{Trace}^{\mathcal{D}}_{\mathsf{PS}}(\mathsf{PP}, \mathbb{A}_{\mathcal{D}}, \epsilon) \to \mathbb{K}_T \subseteq \{1, \ldots, \mathcal{K}\}$. $\mathsf{Trace}_{\mathsf{PS}}$ *is an oracle algorithm that interacts with a policy-specific decryption blackbox $\mathcal{D}$. By given the public parameter* PP*, an access policy $\mathbb{A}_{\mathcal{D}}$, and a probability value ϵ, the algorithm runs in time polynomial in λ and $1/\epsilon$, and outputs an index set $\mathbb{K}_T \subseteq \{1, \ldots, \mathcal{K}\}$ which identifies the set of malicious users. Note that ϵ has to be polynomially related to λ.*

The following $\mathsf{Tracing\ Game}$ captures the notion of fully collusion-resistant traceability against policy-specific decryption blackbox.

$\mathsf{Game}^{\mathsf{PS}}_{\mathsf{TR}}$. The $\mathsf{Tracing\ Game}$ is defined between a challenger and an adversary $\mathcal{A}$ as follows:

- *Setup:* The challenger runs $\mathsf{Setup}(\lambda, \mathcal{U}, \mathcal{K})$ and gives the public parameter PP to $\mathcal{A}$.
- *Key Query:* For $i = 1$ to q, $\mathcal{A}$ adaptively submits (k_i, S_{k_i}), and gets $\mathsf{SK}_{k_i, S_{k_i}}$ from the challenger.
- *(Policy-Specific) Decryption Blackbox Generation:* $\mathcal{A}$ outputs a decryption blackbox $\mathcal{D}$ associated with an access policy $\mathbb{A}_{\mathcal{D}}$ and a non-negligible probability value ϵ.
- *Tracing:* The challenger runs $\mathsf{Trace}^{\mathcal{D}}_{\mathsf{PS}}(\mathsf{PP}, \mathbb{A}_{\mathcal{D}}, \epsilon)$ to obtain an index set $\mathbb{K}_T \subseteq \{1, \ldots, \mathcal{K}\}$.

Let $\mathbb{K}_{\mathcal{D}} = \{k_i | 1 \leq i \leq q\}$ be the index set of keys corrupted by the adversary. We say that the adversary $\mathcal{A}$ wins the game if the following conditions hold:

1) $\Pr[\mathcal{D}(\mathsf{Encrypt}(\mathsf{PP}, M, \mathbb{A}_{\mathcal{D}})) = M] \geq \epsilon$, where the probability is taken over the random choices of message M and the random coins of $\mathcal{D}$. A decryption blackbox satisfying this condition is said to be a *useful policy-specific decryption blackbox*.
2) $\mathbb{K}_T = \emptyset$, or $\mathbb{K}_T \not\subseteq \mathbb{K}_{\mathcal{D}}$, or ($S_{k_t}$ *does not satisfy* $\mathbb{A}_{\mathcal{D}}$ $\forall k_t \in \mathbb{K}_T$).

We denote by $\mathsf{TRAdv}^{\mathsf{PS}}_{\mathcal{A}}$ the probability that adversary $\mathcal{A}$ wins this game.

Remark: For a useful policy-specific decryption blackbox $\mathcal{D}$, the traced $\mathbb{K}_T$ must satisfy $(\mathbb{K}_T \neq \emptyset) \wedge (\mathbb{K}_T \subseteq \mathbb{K}_{\mathcal{D}}) \wedge (\exists k_t \in \mathbb{K}_T\ s.t.\ S_{k_t}\ satisfies\ \mathbb{A}_{\mathcal{D}})$. (1) $(\mathbb{K}_T \neq \emptyset) \wedge (\mathbb{K}_T \subseteq \mathbb{K}_{\mathcal{D}})$ captures the preliminary traceability that $\mathsf{Trace}_{\mathsf{PS}}$ can extract at least one malicious user and the coalition of malicious users cannot frame any innocent user. (2) $(\exists k_t \in \mathbb{K}_T\ s.t.\ S_{k_t}\ satisfies\ \mathbb{A}_{\mathcal{D}})$ captures the *strong traceability* that $\mathsf{Trace}_{\mathsf{PS}}$ can extract at least one malicious user whose attribute set can satisfy the access policy $\mathbb{A}_{\mathcal{D}}$.

Definition 3: A $\mathcal{K}$-user CP-ABE system is traceable against policy-specific decryption blackbox if for all polynomial-time adversaries $\mathcal{A}$ the advantage $\mathsf{TRAdv}^{\mathsf{PS}}_{\mathcal{A}}$ *is negligible in λ.*

We say that a $\mathcal{K}$-user CP-ABE system is *selectively* traceable against policy-specific decryption blackbox if we add an **Init** stage before **Setup** for the adversary to commit the access policy $\mathbb{A}_{\mathcal{D}}$.

D. Traceability Against Policy-Specific Decryption Blackbox Implies Traceability Against Key-Like Decryption Blackbox

From the definitions in previous two sections (Sec. II-B and Sec. II-C), we can see that key-like decryption blackbox is often more powerful than policy-specific decryption blackbox, since a key-like decryption blackbox associated with attribute set $S_{\mathcal{D}}$ can decrypt the ciphertexts *as long as the access policy is satisfied by $S_{\mathcal{D}}$*, while a policy-specific decryption blackbox can decrypt only the ciphertexts generated under a specific access policy. As a result, intuitively, tracing a key-like decryption blackbox might be easier than tracing a policy-specific decryption blackbox. In the following theorem, we formally prove that *for CP-ABE's traceability, investigating traceability against policy-specific decryption blackbox is sufficient.*

Theorem 1: For a secure CP-ABE scheme, if it is (selectively) traceable against policy-specific decryption blackbox then it is also (selectively) traceable against key-like decryption blackbox.

Proof: Suppose Σ is a secure CP-ABE scheme with (selective) traceability against policy-specific decryption blackbox and the tracing algorithm is $\mathsf{Trace}_{\mathsf{PS}}$. Consider any useful key-like decryption blackbox $\mathcal{D}$, let $S_{\mathcal{D}}$ and ϵ denote the associated non-empty attribute set and non-negligible probability value respectively. Let $\mathbb{A}_{S_{\mathcal{D}}} = \bigwedge_{x \in S_{\mathcal{D}}} x$, we have that $\mathcal{D}$ can decrypt the ciphertexts generated under $\mathbb{A}_{S_{\mathcal{D}}}$ with probability at least ϵ, since $\mathcal{D}$ is useful and $\mathbb{A}_{S_{\mathcal{D}}}$ is satisfied by $S_{\mathcal{D}}$. Thus we can regard $\mathcal{D}$ as a policy-specific decryption blackbox associated with access policy $\mathbb{A}_{S_{\mathcal{D}}}$ and non-negligible value ϵ. We construct the tracing algorithm $\mathsf{Trace}_{\mathsf{KL}}$ as

$$\mathsf{Trace}^{\mathcal{D}}_{\mathsf{KL}}(\mathsf{PP}, S_{\mathcal{D}}, \epsilon) = \mathsf{Trace}^{\mathcal{D}}_{\mathsf{PS}}(\mathsf{PP}, \mathbb{A}_{S_{\mathcal{D}}}, \epsilon).$$

Let $\mathbb{K}_{\mathcal{D}}$ be the index set of keys corrupted by the adversary, and $\mathbb{K}_T$ be the index set output by $\mathsf{Trace}^{\mathcal{D}}_{\mathsf{KL}}(\mathsf{PP}, S_{\mathcal{D}}, \epsilon)$ (i.e. output by $\mathsf{Trace}^{\mathcal{D}}_{\mathsf{PS}}(\mathsf{PP}, \mathbb{A}_{S_{\mathcal{D}}}, \epsilon)$). As $\mathcal{D}$ is (selectively) traceable against policy-specific decryption blackbox, we have that

$$(\mathbb{K}_T \neq \emptyset) \wedge (\mathbb{K}_T \subseteq \mathbb{K}_{\mathcal{D}}) \\ \wedge (\exists k_t \in \mathbb{K}_T\ s.t.\ S_{k_t}\ satisfies\ \mathbb{A}_{S_{\mathcal{D}}}).$$

[1] In fact, the Trace algorithm against key-like decryption blackbox in [15] makes use of a powerful function in key-like blackbox, i.e. to trace a key-like blackbox with attribute set $S_{\mathcal{D}}$, the Trace algorithm feeds the blackbox with ciphertexts generated under $\mathbb{A}_{\mathcal{D}} = \bigwedge_{x \in S_{\mathcal{D}}} x$, which is a limited policy but is still satisfied by $S_{\mathcal{D}}$.

As $\mathbb{A}_{S_{\mathcal{D}}} = \bigwedge_{x \in S_{\mathcal{D}}} x$, we have that "$S_{k_t}$ *satisfies* $\mathbb{A}_{S_{\mathcal{D}}}$" holds if and only if "$S_{k_t} \supseteq S_{\mathcal{D}}$". Thus,

$$(\mathbb{K}_T \neq \emptyset) \wedge (\mathbb{K}_T \subseteq \mathbb{K}_{\mathcal{D}}) \wedge (\exists k_t \in \mathbb{K}_T \ s.t. \ S_{k_t} \supseteq S_{\mathcal{D}}).$$

This implies that Σ, equipped with this tracing algorithm $\mathsf{Trace}_{\mathsf{KL}}$ (which is constructed from $\mathsf{Trace}_{\mathsf{PS}}$), is (selectively) traceable against key-like decryption blackbox. □

As an evidence that achieving traceability against key-like decryption blackbox is easier than that against policy-specific decryption blackbox, while in [15] we already proved that the proposed CP-ABE scheme is *adaptively* traceable against key-like decryption blackbox, in this extended version we show that it is only *selectively* traceable against policy-specific decryption blackbox.

III. Augmented CP-ABE

In this section, we redefine the index-hiding property, which is more general and stronger than that in [15]. We then show that an Augmented CP-ABE (or AugCP-ABE for short), with its definition reviewed below, can be transformed into a traceable CP-ABE against policy-specific decryption blackbox provided that message-hiding and the newly defined index-hiding are satisfied.

A. Definitions

An AugCP-ABE system consists of the following four algorithms, in particular, different from CP-ABE, the encryption algorithm takes one more parameter $\bar{k} \in \{1, \ldots, \mathcal{K}+1\}$.

- $\mathsf{Setup}_{\mathsf{A}}(\lambda, \mathcal{U}, \mathcal{K}) \to (\mathsf{PP}, \mathsf{MSK})$. The algorithm takes as input a security parameter λ, the attribute universe $\mathcal{U}$, and the number of users $\mathcal{K}$ in the system, then runs in polynomial time in λ, and outputs the public parameter PP and a master secret key MSK.
- $\mathsf{KeyGen}_{\mathsf{A}}(\mathsf{PP}, \mathsf{MSK}, S) \to \mathsf{SK}_{k,S}$. The algorithm takes as input PP, MSK, and an attribute set S, and outputs a private key $\mathsf{SK}_{k,S}$, which is assigned with a unique index $k \in \{1, \ldots, \mathcal{K}\}$.
- $\mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, \mathbb{A}, \bar{k}) \to CT$. The algorithm takes as input PP, a message M, an access policy $\mathbb{A}$ over $\mathcal{U}$, and an index $\bar{k} \in \{1, \ldots, \mathcal{K}+1\}$, and outputs a ciphertext CT. $\mathbb{A}$ **is included in** CT**, but the value of** $\bar{k}$ **is not**.
- $\mathsf{Decrypt}_{\mathsf{A}}(\mathsf{PP}, CT, \mathsf{SK}_{k,S}) \to M$ or $\perp$. The algorithm takes as input PP, CT, and $\mathsf{SK}_{k,S}$. If S satisfies CT's access policy, the algorithm outputs a message M, otherwise it outputs $\perp$.

Correctness. For any attribute set $S \subseteq \mathcal{U}$, $k \in \{1, \ldots, \mathcal{K}\}$, access policy $\mathbb{A}$ over $\mathcal{U}$, $\bar{k} \in \{1, \ldots, \mathcal{K}+1\}$, and message M, suppose $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}_{\mathsf{A}}(\lambda, \mathcal{U}, \mathcal{K})$, $\mathsf{SK}_{k,S} \leftarrow \mathsf{KeyGen}_{\mathsf{A}}(\mathsf{PP}, \mathsf{MSK}, S)$, $CT \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, \mathbb{A}, \bar{k})$. If ($S$ satisfies $\mathbb{A}$) $\wedge (k \geq \bar{k})$ then $\mathsf{Decrypt}_{\mathsf{A}}(\mathsf{PP}, CT, \mathsf{SK}_{k,S}) = M$.

It is worth noticing that during decryption if the S of a private decryption key satisfies the access policy $\mathbb{A}$ of a ciphertext, $\mathsf{Decrypt}_{\mathsf{A}}$ will output a message, but whether the output message is equal to the encrypted message is determined by the relation of k and $\bar{k}$. i.e., if and only if (S satisfies $\mathbb{A}$) $\wedge$ $(k \geq \bar{k})$, can $\mathsf{SK}_{k,S}$ correctly decrypt a ciphertext encrypted using $(\mathbb{A}, \bar{k})$. Note that if we set $\bar{k} = 1$, then the functions of AugCP-ABE are identical to that of CP-ABE.

Security. We define the security of AugCP-ABE in the following three games, where the first two are for message-hiding, and the third one is for the index-hiding property. In the first two **message-hiding games** between a challenger and an adversary $\mathcal{A}$, $\bar{k} = 1$ (the first game, $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$) and $\bar{k} = \mathcal{K}+1$ (the second game, $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_{\mathcal{K}+1}}$).

- *Setup:* The challenger runs $\mathsf{Setup}_{\mathsf{A}}(\lambda, \mathcal{U}, \mathcal{K})$ and gives the public parameter PP to $\mathcal{A}$.
- *Phase 1:* For $i = 1$ to q_1, $\mathcal{A}$ adaptively submits (k_i, S_{k_i}), and the challenger responds with $\mathsf{SK}_{k_i, S_{k_i}}$.
- *Challenge:* $\mathcal{A}$ submits two equal-length messages M_0, M_1 and an access policy $\mathbb{A}^*$. The challenger flips a random coin $b \in \{0, 1\}$, and sends $CT \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M_b, \mathbb{A}^*, \bar{k})$ to $\mathcal{A}$.
- *Phase 2:* For $i = q_1 + 1$ to q, $\mathcal{A}$ adaptively submits (k_i, S_{k_i}), and gets $\mathsf{SK}_{k_i, S_{k_i}}$ from the challenger.
- *Guess:* $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ for b.

$\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$. In the Challenge phase the challenger sends $CT \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M_b, \mathbb{A}^*, 1)$ to $\mathcal{A}$. $\mathcal{A}$ wins the game if $b' = b$ under the **restriction** that $\mathbb{A}^*$ cannot be satisfied by any of the queried attribute sets $S_{k_1}, \ldots, S_{k_q}$. The advantage of $\mathcal{A}$ is defined as $\mathsf{MH}^{\mathsf{A}}_1\mathsf{Adv}_{\mathcal{A}} = |\Pr[b' = b] - \frac{1}{2}|$.

$\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_{\mathcal{K}+1}}$. In the Challenge phase the challenger sends $CT \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M_b, \mathbb{A}^*, \mathcal{K}+1)$ to $\mathcal{A}$. $\mathcal{A}$ wins the game if $b' = b$. The advantage of $\mathcal{A}$ is defined as $\mathsf{MH}^{\mathsf{A}}_{\mathcal{K}+1}\mathsf{Adv}_{\mathcal{A}} = |\Pr[b' = b] - \frac{1}{2}|$.

Definition 4: A $\mathcal{K}$-user Augmented CP-ABE system is message-hiding if for all polynomial-time adversaries $\mathcal{A}$ the advantages $\mathsf{MH}^{\mathsf{A}}_1\mathsf{Adv}_{\mathcal{A}}$ and $\mathsf{MH}^{\mathsf{A}}_{\mathcal{K}+1}\mathsf{Adv}_{\mathcal{A}}$ are negligible in λ.

$\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$. In the third game, **index-hiding game**, we require that, for any access policy $\mathbb{A}^*$, an adversary cannot distinguish between an encryption using $(\mathbb{A}^*, \bar{k})$ and $(\mathbb{A}^*, \bar{k}+1)$ without a private decryption key $\mathsf{SK}_{\bar{k}, S_{\bar{k}}}$ where $S_{\bar{k}}$ *satisfies* $\mathbb{A}^*$. The game takes as input a parameter $\bar{k} \in \{1, \ldots, \mathcal{K}\}$ which is given to both the challenger and the adversary $\mathcal{A}$. The game proceeds as follows:

- *Setup:* The challenger runs $\mathsf{Setup}_{\mathsf{A}}(\lambda, \mathcal{U}, \mathcal{K})$ and gives the public parameter PP to $\mathcal{A}$.
- *Key Query:* For $i = 1$ to q, $\mathcal{A}$ adaptively submits (k_i, S_{k_i}), and gets $\mathsf{SK}_{k_i, S_{k_i}}$ from the challenger.
- *Challenge:* $\mathcal{A}$ submits a message M and an access policy $\mathbb{A}^*$. The challenger flips a random coin $b \in \{0, 1\}$, and sends $CT \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, \mathbb{A}^*, \bar{k}+b)$ to $\mathcal{A}$.
- *Guess:* $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ for b.

$\mathcal{A}$ wins the game if $b' = b$ under the **restriction** that none of the queried pairs $\{(k_i, S_{k_i})\}_{1 \leq i \leq q}$ can satisfy $(k_i = \bar{k}) \wedge$ (S_{k_i} satisfies $\mathbb{A}^*$). The advantage of $\mathcal{A}$ is defined as $\mathsf{IH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}] = |\Pr[b' = b] - \frac{1}{2}|$.

Definition 5: A $\mathcal{K}$-user Augmented CP-ABE system is index-hiding if for all polynomial-time adversaries $\mathcal{A}$ the advantages $\mathsf{IH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}]$ for $\bar{k} = 1, \ldots, \mathcal{K}$ are negligible in λ.

We say that an Augmented CP-ABE system is *selectively* index-hiding if we add an **Init** stage before **Setup** for the adversary to commit the challenge access policy $\mathbb{A}^*$.

Remark: The index-hiding property defined in [15] is weaker than that defined as above, since in the **Challenge** phase of the index hiding game of [15], $\mathcal{A}$ submits a message M and a non-empty attribute set S^*, then the challenger flips a random coin $b \in \{0,1\}$, and sends $CT \leftarrow \mathsf{Encrypt_A}(\mathsf{PP}, M, \mathbb{A}_{S^*}, \bar{k} + b)$ to $\mathcal{A}$, where $\mathbb{A}_{S^*} = \bigwedge_{x \in S^*} x$. Hence, the index-hiding property in [15] is limited to a specific type of access policy, AND policy. *In this paper we refer to the index-hiding property defined in [15] as* ***weak index-hiding****, and the corresponding game as* ***weak index-hiding game***. While [15] shows that such a weak index-hiding property is sufficient for achieving traceability against key-like decryption blackbox, in the next section, we show that the new index-hiding property defined above can achieve traceability against policy-specific decryption blackbox.

B. Reducing CP-ABE With Traceability to AugCP-ABE

Let $\Sigma_{\mathsf{A}} = (\mathsf{Setup_A}, \mathsf{KeyGen_A}, \mathsf{Encrypt_A}, \mathsf{Decrypt_A})$ be an AugCP-ABE, define $\mathsf{Encrypt}(\mathsf{PP}, M, \mathbb{A}) = \mathsf{Encrypt_A}(\mathsf{PP}, M, \mathbb{A}, 1)$, then $\Sigma = (\mathsf{Setup_A}, \mathsf{KeyGen_A}, \mathsf{Encrypt}, \mathsf{Decrypt_A})$ is a CP-ABE derived from Σ_{A}. In the following, we construct a tracing algorithm $\mathsf{Trace_{PS}}$ for Σ and show that if Σ_{A} is message-hiding and index-hiding (resp. selectively index-hiding), then Σ (equipped with $\mathsf{Trace_{PS}}$) is a secure CP-ABE with traceability (resp. selective traceability) against policy-specific decryption blackbox.

$\mathsf{Trace}^{\mathcal{D}}_{\mathsf{PS}}(\mathsf{PP}, \mathbb{A}_{\mathcal{D}}, \epsilon) \rightarrow \mathbb{K}_T \subseteq \{1, \ldots, \mathcal{K}\}$. Given a policy-specific decryption blackbox $\mathcal{D}$ associated with an access policy $\mathbb{A}_{\mathcal{D}}$ and probability $\epsilon > 0$, the tracing algorithm works as follows: [2]

1) For $k = 1$ to $\mathcal{K} + 1$, do the following:
   a) The algorithm repeats the following $8\lambda(\mathcal{K}/\epsilon)^2$ times:
      i) Sample M from the message space at random.
      ii) Let $CT \leftarrow \mathsf{Encrypt_A}(\mathsf{PP}, M, \mathbb{A}_{\mathcal{D}}, k)$.
      iii) Query oracle $\mathcal{D}$ on input CT which contains $\mathbb{A}_{\mathcal{D}}$, and compare the output of $\mathcal{D}$ with M.
   b) Let $\hat{p}_k$ be the fraction of times that $\mathcal{D}$ decrypted the ciphertexts correctly.
2) Let $\mathbb{K}_T$ be the set of all $k \in \{1, \ldots, \mathcal{K}\}$ for which $\hat{p}_k - \hat{p}_{k+1} \geq \epsilon/(4\mathcal{K})$. Then output $\mathbb{K}_T$ as the index set of the private decryption keys of malicious users.

Theorem 2: If Σ_{A} is an AugCP-ABE with message-hiding and index-hiding (resp. selective index-hiding) properties, then Σ is a secure CP-ABE with traceability (resp. selective traceability) against policy-specific decryption blackbox.

Proof: Note that the tracing algorithm $\mathsf{Trace_{PS}}$ above is same to the tracing algorithm Trace in [15], which is design to trace key-like decryption blackbox associated with attribute set $S_{\mathcal{D}}$, except that the ciphertexts are generated under $\mathbb{A}_{\mathcal{D}}$ rather than $\mathbb{A}_{S_{\mathcal{D}}} = \bigwedge_{x \in S_{\mathcal{D}}} x$. The proof of this theorem is identical to that of [15, Theorem 1], replacing "$S_{k_t} \supseteq S_{\mathcal{D}}$" with "$S_{k_t}$ *satisfies* $\mathbb{A}_{\mathcal{D}}$". □

IV. An Efficient Augmented CP-ABE

In [15] we proposed an efficient and expressive AugCP-ABE construction, and proved that it satisfies message-hiding, which implies that the derived CP-ABE is fully secure. Also we proved that the construction is *weak* index-hiding, which implies that the derived CP-ABE is adaptively traceable against key-like decryption blackbox. In this section, we prove that the AugCP-ABE satisfies *selective* index-hiding, which implies that the derived CP-ABE is selectively traceable against policy-specific decryption blackbox. We first review the AugCP-ABE construction below.

A. Preliminaries

1) Linear Secret-Sharing Schemes (LSSS): An LSSS is a share-generating matrix A whose rows $\{A_i\}$ are labeled by attributes through a function ρ. When we consider a 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, $A\vec{v}$ is the vector of l shares of the secret s, and the share $\lambda_i = (A\vec{v})_i$, i.e. the inner product $A_i \cdot \vec{v}$, belongs to attribute $\rho(i)$. An attribute set S satisfies the LSSS access matrix if the rows labeled by the attributes in S have the *linear reconstruction* property, which means that there exist constants $\{\omega_i\}$ such that, for any valid shares $\{\lambda_i\}$ of a secret s according to the LSSS matrix, we have $\sum_i \omega_i \lambda_i = s$. Essentially, a user will be able to decrypt a ciphertext with access matrix A if and only if the rows of A labeled by the user's attributes include the vector $(1, 0, \ldots, 0)$ in their span.

2) Composite Order Bilinear Groups: Let $\mathcal{G}$ be a group generator, which takes a security parameter λ and outputs $(p_1, p_2, p_3, \mathbb{G}, \mathbb{G}_T, e)$ where p_1, p_2, p_3 are distinct primes, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order $N = p_1 p_2 p_3$, and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ a map such that: (1) (Bilinear) $\forall g, h \in \mathbb{G}, a, b \in \mathbb{Z}_N, e(g^a, h^b) = e(g,h)^{ab}$, (2) (Non-Degenerate) $\exists g \in \mathbb{G}$ such that $e(g,g)$ has order N in $\mathbb{G}_T$. Assume that group operations in $\mathbb{G}$ and $\mathbb{G}_T$ as well as the bilinear map e are computable in polynomial time with respect to λ. Let $\mathbb{G}_{p_1}$, $\mathbb{G}_{p_2}$ and $\mathbb{G}_{p_3}$ be the subgroups of order p_1, p_2 and p_3 in $\mathbb{G}$, respectively. These subgroups are "orthogonal" to each other under the bilinear map e: if $h_i \in \mathbb{G}_{p_i}$ and $h_j \in \mathbb{G}_{p_j}$ for $i \neq j$, then $e(h_i, h_j) = 1$ (the identity element in $\mathbb{G}_T$).

3) Complexity Assumptions: The message-hiding property of our AugCP-ABE construction relies on four assumptions (the Assumption 1 in [12], the General Subgroup Decision Assumption, the 3-Party Diffie-Hellman Assumption in a Subgroup, and the Source Group q-Parallel BDHE Assumption in a Subgroup). While in [15] we proved that the weak index-hiding property relies on the 3-Party Diffie-Hellman Assumption and Decisional Linear Assumption, in this paper, to prove the more challenging index-hiding property, we

[2] The tracing algorithm uses a technique based on that in broadcast encryption by [16]–[18].

need to introduce a new assumption, which is modified from the Source Group q-Parallel BDHE Assumption in a Subgroup [12] and we refer to it as the Modified Source Group q-Parallel BDHE Assumption in a Subgroup. Here we only review this new assumption, and refer to [12] and [18] for the details of the other assumptions. From now on, for a positive integer, for example q, we use the notation $[q]$ to denote the set $\{1, 2, \ldots, q\}$.

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

$$\begin{aligned}
&(N = p_1 p_2 p_3, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G},\\
&g \xleftarrow{R} \mathbb{G}_{p_1}, \quad g_2 \xleftarrow{R} \mathbb{G}_{p_2}, \quad g_3 \xleftarrow{R} \mathbb{G}_{p_3},\\
&c, d, f, b_1, \ldots, b_q \xleftarrow{R} \mathbb{Z}_N,\\
D = \Big(&(N, \mathbb{G}, \mathbb{G}_T, e), g, g_2, g_3, g^f, g^{df}, g^{fc^q}, g^c, g^{c^2}, \ldots, g^{c^q},\\
&g^{c^i/b_j} \quad \forall i \in [2q] \setminus \{q+1\}, j \in [q],\\
&g^{dfb_j} \quad \forall j \in [q],\\
&g^{dfc^i b_{j'}/b_j} \quad \forall i \in [q], j, j' \in [q] \ s.t. \ j \neq j' \Big),\\
&T_0 = g^{dc^{q+1}}, T_1 \xleftarrow{R} \mathbb{G}_{p_1}.
\end{aligned}$$

The advantage of an algorithm $\mathcal{A}$ in breaking the Modified Source Group q-Parallel BDHE Assumption in a Subgroup is:

$$Adv_{\mathcal{G},\mathcal{A}}^{qMPBDHE}(\lambda) := |\Pr[\mathcal{A}(D, T_0) = 1] - \Pr[\mathcal{A}(D, T_1) = 1]|.$$

Definition 6: $\mathcal{G}$ satisfies the Modified Source Group q-Parallel BDHE Assumption in a Subgroup if $Adv_{\mathcal{G},\mathcal{A}}^{qMPBDHE}(\lambda)$ is a negligible function of λ for any polynomial time algorithm $\mathcal{A}$.

Compared with the original Source Group q-Parallel BDHE Assumption in a Subgroup in [12], the adversary in the assumption above is given one more element g^{fc^q}, but is not given $g^{c^{q+2}}, \ldots, g^{c^{2q}}$. In Appendix B, we prove that the prime order variant of the above assumption holds in the generic group model (the proof for this version follows analogously).

Notations. Suppose the number of users $\mathcal{K}$ in the system equals m^2 for some m[3]. We arrange the users in an $m \times m$ matrix and uniquely assign a tuple (i, j) where $1 \leq i, j \leq m$, to each user. A user at position (i, j) of the matrix has index $k = (i-1) * m + j$. For simplicity, we directly use (i, j) as the index where $(i, j) \geq (\bar{i}, \bar{j})$ means that $((i > \bar{i}) \vee (i = \bar{i} \wedge j \geq \bar{j}))$. The use of pairwise notation (i, j) is purely a notational convenience, as $k = (i-1) * m + j$ defines a bijection between $\{(i, j) | 1 \leq i, j \leq m\}$ and $\{1, \ldots, \mathcal{K}\}$. For a given vector $\vec{v} = (v_1, \ldots, v_d)$, by $g^{\vec{v}}$ we mean the vector $(g^{v_1}, \ldots, g^{v_d})$. Furthermore, for $g^{\vec{v}} = (g^{v_1}, \ldots, g^{v_d})$ and $g^{\vec{w}} = (g^{w_1}, \ldots, g^{w_d})$, by $g^{\vec{v}} \cdot g^{\vec{w}}$ we mean the vector $(g^{v_1+w_1}, \ldots, g^{v_d+w_d})$, i.e. $g^{\vec{v}} \cdot g^{\vec{w}} = g^{\vec{v}+\vec{w}}$, and by $e_d(g^{\vec{v}}, g^{\vec{w}})$ we mean $\prod_{k=1}^{d} e(g^{v_k}, g^{w_k})$, i.e. $e_d(g^{\vec{v}}, g^{\vec{w}}) = \prod_{k=1}^{d} e(g^{v_k}, g^{w_k}) = e(g, g)^{(\vec{v} \cdot \vec{w})}$ where $(\vec{v} \cdot \vec{w})$ is the inner product of $\vec{v}$ and $\vec{w}$. Given a bilinear group order N, one can randomly choose $r_x, r_y, r_z \in \mathbb{Z}_N$, and set $\vec{\chi}_1 = (r_x, 0, r_z)$, $\vec{\chi}_2 = (0, r_y, r_z)$, $\vec{\chi}_3 = \vec{\chi}_1 \times \vec{\chi}_2 = (-r_y r_z, -r_x r_z, r_x r_y)$. Let $span\{\vec{\chi}_1, \vec{\chi}_2\}$ be the subspace spanned by $\vec{\chi}_1$ and $\vec{\chi}_2$, i.e. $span\{\vec{\chi}_1, \vec{\chi}_2\} = \{\nu_1 \vec{\chi}_1 + \nu_2 \vec{\chi}_2 | \nu_1, \nu_2 \in \mathbb{Z}_N\}$. We can see that $\vec{\chi}_3$ is orthogonal to the subspace $span\{\vec{\chi}_1, \vec{\chi}_2\}$ and $\mathbb{Z}_N^3 = span\{\vec{\chi}_1, \vec{\chi}_2, \vec{\chi}_3\} = \{\nu_1 \vec{\chi}_1 + \nu_2 \vec{\chi}_2 + \nu_3 \vec{\chi}_3 | \nu_1, \nu_2, \nu_3 \in \mathbb{Z}_N\}$. For any $\vec{v} \in span\{\vec{\chi}_1, \vec{\chi}_2\}$, we have $(\vec{\chi}_3 \cdot \vec{v}) = 0$, and for random $\vec{v} \in \mathbb{Z}_N^3$, $(\vec{\chi}_3 \cdot \vec{v}) \neq 0$ happens with overwhelming probability.

B. AugCP-ABE Construction

Now we briefly review the AugCP-ABE construction of [15] as follows.

- $\mathsf{Setup_A}(\lambda, \mathcal{U}, \mathcal{K} = m^2) \rightarrow (\mathsf{PP}, \mathsf{MSK})$. Let $\mathbb{G}$ be a bilinear group of order $N = p_1 p_2 p_3$ (3 distinct primes, whose size is determined by λ), $\mathbb{G}_{p_i}$ the subgroup of order p_i in $\mathbb{G}$ (for $i = 1, 2, 3$), and $g, f, h \in \mathbb{G}_{p_1}$, $g_3 \in \mathbb{G}_{p_3}$ the generators of corresponding subgroups. The algorithm randomly chooses exponents $\{\alpha_i, \ r_i, \ z_i \in \mathbb{Z}_N\}_{i \in [m]}$, $\{c_j \in \mathbb{Z}_N\}_{j \in [m]}$, $\{a_x \in \mathbb{Z}_N\}_{x \in \mathcal{U}}$. The public parameter PP includes the description of the group and the following elements:

  $$\big(g, \ f, \ h, \{E_i = e(g, g)^{\alpha_i}, G_i = g^{r_i}, \ Z_i = g^{z_i}\}_{i \in [m]}, \{H_j = g^{c_j}\}_{j \in [m]}, \ \{U_x = g^{a_x}\}_{x \in \mathcal{U}}\big).$$

  The master secret key is set to

  $$\mathsf{MSK} = (\alpha_1, \ldots, \alpha_m, \ \ r_1, \ldots, r_m, \ \ c_1, \ldots, c_m, \ \ g_3).$$

  A counter $ctr = 0$ is implicitly included in MSK.
- $\mathsf{KeyGen_A}(\mathsf{PP}, \mathsf{MSK}, S) \rightarrow \mathsf{SK}_{(i,j),S}$. The algorithm first sets $ctr = ctr + 1$ and computes the corresponding index in the form of (i, j) where $1 \leq i, j \leq m$ and $(i-1) * m + j = ctr$. Then it randomly chooses $\sigma_{i,j}, \delta_{i,j} \in \mathbb{Z}_N$ and $R, R', R'', R''', R_x (x \in S) \in \mathbb{G}_{p_3}$, and outputs a private key $\mathsf{SK}_{(i,j),S} = \big((i, j), S, K_{i,j}, K'_{i,j}, K''_{i,j}, K'''_{i,j}, \{K_{i,j,x}\}_{x \in S}\big)$ as

  $$\begin{aligned}
  K_{i,j} &= g^{\alpha_i} g^{r_i c_j} f^{\sigma_{i,j}} h^{\delta_{i,j}} R,\\
  K'_{i,j} &= g^{\sigma_{i,j}} R', \ \ K''_{i,j} = g^{\delta_{i,j}} R'', \ \ K'''_{i,j} = Z_i^{\sigma_{i,j}} R''',\\
  K_{i,j,x} &= U_x^{\sigma_{i,j}} R_x \ \ \forall x \in S.
  \end{aligned}$$
- $\mathsf{Encrypt_A}(\mathsf{PP}, M, \mathbb{A} = (A, \rho), (\bar{i}, \bar{j})) \rightarrow CT$. A is an $l \times n$ LSSS matrix and ρ maps each row A_k of A to an attribute $\rho(k) \in \mathcal{U}$. The algorithm randomly chooses

  $$\begin{aligned}
  &\kappa, \ \tau, \ \ s_1, \ldots, s_m, \ \ t_1, \ldots, t_m \ \in \mathbb{Z}_N,\\
  &\vec{v}_c, \ \vec{w}_1, \ldots, \vec{w}_m \ \in \mathbb{Z}_N^3,\\
  &\xi_1, \ldots, \xi_l \ \in \mathbb{Z}_N, \quad \vec{u} = (\pi, u_2, \ldots, u_n) \ \in \mathbb{Z}_N^n.
  \end{aligned}$$

  In addition, it randomly chooses $r_x, r_y, r_z \in \mathbb{Z}_N$, and sets $\vec{\chi}_1 = (r_x, 0, r_z)$, $\vec{\chi}_2 = (0, r_y, r_z)$, $\vec{\chi}_3 = \vec{\chi}_1 \times \vec{\chi}_2 = (-r_y r_z, -r_x r_z, r_x r_y)$. Then it randomly chooses

  $$\begin{aligned}
  &\vec{v}_i \in \mathbb{Z}_N^3 \quad \forall i \in \{1, \ldots, \bar{i}\},\\
  &\vec{v}_i \in span\{\vec{\chi}_1, \vec{\chi}_2\} \quad \forall i \in \{\bar{i}+1, \ldots, m\},
  \end{aligned}$$

  and creates the ciphertext $\langle (A, \rho), \ (\vec{R}_i, \vec{R}'_i, Q_i, Q'_i, Q''_i, Q'''_i, T_i)_{i=1}^{m}, (\vec{C}_j, \vec{C}'_j)_{j=1}^{m}, \ (P_k, P'_k)_{k=1}^{l} \rangle$ as follows:

[3] If the number of users is not a square, we add some “dummy” users to pad to the next square.

1) For each row $i \in [m]$:
   - if $i < \bar{i}$: randomly chooses $\hat{s}_i \in \mathbb{Z}_N$, and sets
     $$\vec{R}_i = g^{\vec{v}_i},\ \ \vec{R}'_i = g^{\kappa\vec{v}_i},$$
     $$Q_i = g^{s_i}, Q'_i = f^{s_i} Z_i^{t_i} f^{\pi}, Q''_i = h^{s_i}, Q'''_i = g^{t_i},$$
     $$T_i = E_i^{\hat{s}_i}.$$
   - if $i \geq \bar{i}$: sets
     $$\vec{R}_i = G_i^{s_i\vec{v}_i},\quad \vec{R}'_i = G_i^{\kappa s_i\vec{v}_i},$$
     $$Q_i = g^{\tau s_i(\vec{v}_i\cdot\vec{v}_c)}, Q'_i = f^{\tau s_i(\vec{v}_i\cdot\vec{v}_c)} Z_i^{t_i} f^{\pi},$$
     $$Q''_i = h^{\tau s_i(\vec{v}_i\cdot\vec{v}_c)},\quad Q'''_i = g^{t_i},$$
     $$T_i = M \cdot E_i^{\tau s_i(\vec{v}_i\cdot\vec{v}_c)}.$$
2) For each column $j \in [m]$:
   - if $j < \bar{j}$: randomly chooses $\mu_j \in \mathbb{Z}_N$, and sets $\vec{C}_j = H_j^{\tau(\vec{v}_c+\mu_j\vec{\chi}_3)} \cdot g^{\kappa\vec{\omega}_j},\quad \vec{C}'_j = g^{\vec{\omega}_j}$.
   - if $j \geq \bar{j}$: sets $\vec{C}_j = H_j^{\tau\vec{v}_c} \cdot g^{\kappa\vec{\omega}_j},\quad \vec{C}'_j = g^{\vec{\omega}_j}$.
3) For each $k \in [l]$: sets $P_k = f^{A_k\cdot\vec{u}} U_{\rho(k)}^{-\xi_k},\quad P'_k = g^{\xi_k}$.

- $\mathsf{Decrypt_A}(\mathsf{PP}, CT, \mathsf{SK}_{(i,j),S}) \rightarrow M$ or $\perp$. The algorithm parses CT to $\langle(A,\rho), (\vec{R}_i, \vec{R}'_i, Q_i, Q'_i, Q''_i, Q'''_i, T_i)_{i=1}^m, (\vec{C}_j, \vec{C}'_j)_{j=1}^m, (P_k, P'_k)_{k=1}^l\rangle$, and outputs $\perp$ if S does not satisfy (A,ρ), otherwise it
  1) computes constants $\{\omega_k \in \mathbb{Z}_N\}$ such that $\sum_{\rho(k)\in S}\omega_k A_k = (1, 0, \ldots, 0)$, then computes
     $$D_P = \prod_{\rho(k)\in S}\left(e(K'_{i,j}, P_k)e(K_{i,j,\rho(k)}, P'_k)\right)^{\omega_k}$$
     $$= \prod_{\rho(k)\in S}\left(e(g^{\sigma_{i,j}}, f^{A_k\cdot\vec{u}})\right)^{\omega_k} = e(g^{\sigma_{i,j}}, f)^{\pi},$$
  2) computes $D_I = \frac{e(K_{i,j},Q_i)\cdot e(K'''_{i,j},Q'''_i)}{e(K'_{i,j},Q'_i)\cdot e(K''_{i,j},Q''_i)} \cdot \frac{e_3(\vec{R}'_i,\vec{C}'_j)}{e_3(\vec{R}_i,\vec{C}_j)}$,
  3) computes $M' = T_i/(D_P \cdot D_I)$ as the output message. Assume the encrypted message is M and the encryption index is $(\bar{i}, \bar{j})$, it can be verified that only when $(i > \bar{i})$ or $(i = \bar{i} \wedge j \geq \bar{j})$, $M' = M$ will hold. This follows from the facts that for $i > \bar{i}$, we have $(\vec{v}_i \cdot \vec{\chi}_3) = 0$ (since $\vec{v}_i \in span\{\vec{\chi}_1, \vec{\chi}_2\}$), and for $i = \bar{i}$, we have that $(\vec{v}_i \cdot \vec{\chi}_3) \neq 0$ happens with overwhelming probability (since $\vec{v}_i$ is randomly chosen from $\mathbb{Z}_N^3$).

C. AugCP-ABE Security

In [15], we have already shown that the AugCP-ABE above is message-hiding (Theorem 3 and Theorem 4 below), and weak index-hiding. Hence, the derived CP-ABE is traceable against key-like decryption blackbox. We now show that the construction is selectively index-hiding, and thus the derived CP-ABE scheme is selectively traceable against policy-specific decryption blackbox.

Theorem 3: [15] *Under Assumption 1, the General Subgroup Decision Assumption, the 3-Party Diffie-Hellman Assumption in a Subgroup, and the Source Group q-Parallel BDHE Assumption in a Subgroup, no polynomial time adversary can win* $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$ *with non-negligible advantage.*

Theorem 4: [15] *No polynomial time adversary can win* $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_{\mathcal{K}+1}}$ *with non-negligible advantage.*

Theorem 5: Suppose that the Modified Source Group q-Parallel BDHE Assumption in a Subgroup, the 3-Party Diffie-Hellman Assumption and the Decisional Linear Assumption hold. Then no polynomial time adversary can selectively win $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$ *with non-negligible advantage.*

Proof: Theorem 5 follows the following Lemma 1 and Lemma 2 immediately. □

Lemma 1: Suppose that the Modified Source Group q-Parallel BDHE Assumption in a Subgroup holds. Then for $\bar{j} < m$ *no polynomial time adversary can selectively distinguish between an encryption to* $(\bar{i}, \bar{j})$ *and* $(\bar{i}, \bar{j}+1)$ *in* $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$ *with non-negligible advantage, provided that the challenge LSSS matrix's size* $l \times n$ *satisfy* $l, n \leq q$.

Proof: In $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$, the adversary $\mathcal{A}$ will eventually behave in one of two different ways:

- ***Case I.*** In Key Query phase, $\mathcal{A}$ will not submit $((\bar{i}, \bar{j}), S_{(\bar{i},\bar{j})})$ for some attribute set $S_{(\bar{i},\bar{j})}$ to query the corresponding private key. In Challenge phase, $\mathcal{A}$ submits a message M and an access policy $\mathbb{A}^*$. There is not any restriction on $\mathbb{A}^*$.
- ***Case II.*** In Key Query phase, $\mathcal{A}$ will submit $((\bar{i}, \bar{j}), S_{(\bar{i},\bar{j})})$ for some attribute set $S_{(\bar{i},\bar{j})}$ to query the corresponding private key. In Challenge phase, $\mathcal{A}$ submits a message M and an access policy $\mathbb{A}^*$ with the restriction that $S_{(\bar{i},\bar{j})}$ does not satisfy $\mathbb{A}^*$.

Case I Is easy to handle as the adversary will not query a private key with the challenge index $(\bar{i}, \bar{j})$. **Case II** captures the index-hiding requirement in that even if a user has a key with index $(\bar{i}, \bar{j})$ he cannot distinguish between an encryption to $(\mathbb{A}^*, (\bar{i}, \bar{j}))$ and $(\mathbb{A}^*, (\bar{i}, \bar{j}+1))$, if the corresponding attribute set $S_{(\bar{i},\bar{j})}$ does not satisfy $\mathbb{A}^*$. This is the most challenging part of achieving strong traceability. In [15], as only weak index-hiding is needed, i.e. the challenge access policy is $\mathbb{A}^* = \bigwedge_{x\in S^*} x$ for some non-empty attribute set S^*, the restriction that $S_{(\bar{i},\bar{j})}$ does not satisfy $\mathbb{A}^*$ is equivalent to $S^* \setminus S_{(\bar{i},\bar{j})} \neq \emptyset$, so that the simulator can choose a random element to guess if it is in $S^* \setminus S_{(\bar{i},\bar{j})}$. Now, we need to prove the index-hiding property for any challenge access policy $\mathbb{A}^*$, the simulator cannot make a similar guess, and we can prove the index-hiding property against only selective adversaries. The proof details of Lemma 1 can be found in Appendix A. □

Lemma 2: Suppose that the Modified Source Group q-Parallel BDHE Assumption in a Subgroup, the 3-Party Diffie-Hellman Assumption and the Decisional Linear Assumption hold. Then for $1 \leq \bar{i} \leq m$ *no polynomial time adversary can selectively distinguish between an encryption to* $(\bar{i}, m)$ *and* $(\bar{i}+1, 1)$ *in* $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$ *with non-negligible advantage.*

Proof: Similar to the proof of Lemma 6.3 in [18], to prove this lemma we define the following set of hybrid experiment: H_1: Encrypt to $(\bar{i}, \bar{j} = m)$; H_2: Encrypt to $(\bar{i}, \bar{j} = m+1)$; and H_3: Encrypt to $(\bar{i}+1, 1)$. Lemma 2 follows from the following Claim 1 and Claim 2. □

Claim 1: If the Modified Source Group q-Parallel BDHE Assumption in a Subgroup holds, then no polynomial time adversary can selectively distinguish between experiment

H_1 and H_2 with non-negligible advantage, provided that the challenge LSSS matrix's size $l \times n$ satisfy $l, n \leq q$.

Proof: The proof is identical to that of Lemma 1. □

Claim 2: Suppose that the 3-Party Diffie-Hellman Assumption and the Decisional Linear Assumption hold. Then no polynomial time adversary can distinguish between experiments H_2 and H_3 with non-negligible advantage.

Proof: In [15], we gave the proof for the same claim as this one, using a reduction from the AugCP-ABE to the Augmented Broadcast Encryption(AugBE) [18]. As the access policy is not used in the AugBE, the reduction will work in both the index-hiding game (of this paper) and weak index-hiding game (of [15]). We refer to [15] for the proof details of this claim. □

V. Conclusion

In this paper, we modelled CP-ABE's traceability notions reflecting the practical scenarios of key-like decryption blackbox and policy-specific decryption blackbox, and showed that if a secure CP-ABE scheme is (selectively) traceable against policy-specific decryption blackbox then it is also (selectively) traceable against key-like decryption blackbox, hence one can focus on constructing traceable CP-ABE schemes against the policy-specific decryption blackbox. Furthermore, we proved that the CP-ABE construction in [15] is selectively traceable against policy-specific decryption blackbox. In other words, now we have a CP-ABE scheme that is fully secure (against adaptive adversaries in the standard model), highly expressive (in supporting any monotonic access structures), adaptively traceable against key-like decryption blackbox, and selectively traceable against policy-specific decryption blackbox, and for a system with fully collusion-resistant blackbox traceability, the construction is efficient with the sub-linear overhead. We leave it as our future work to construct a scheme with adaptive traceability against policy-specific decryption blackbox.

Note that the current construction works on composite order bilinear groups, which have been well known significantly inefficient when compared with the prime order bilinear groups, a counterpart construction that has the same security, expressivity, traceability and comparable (symbolic) efficiency, but works on prime order bilinear groups is desirable, since such a construction on prime order groups may result in more efficient implementation and better practicality. To obtain such a construction, it is a possible approach to apply the construction idea of [15] to the traitor tracing scheme in [18] and the prime order group CP-ABE scheme in [19], which is a counterpart of the composite order group CP-ABE scheme in [12], from which the current construction's composite order group settings are inherited. The ideas might be applicable, but the concrete construction may be non-trivial, since the construction and proof of [19] are very complicated and somewhat different from that of [12]. We also leave this as one of our future work.

Appendix A
Proof of Lemma 1

Proof: Suppose there exists a polynomial time adversary $\mathcal{A}$ that selectively breaks the index-hiding game with advantage ϵ. We build a PPT algorithm $\mathcal{B}$ to solve a Modified Source Group q-parallel BDHE problem instance in a subgroup as follows. $\mathcal{B}$ is given T and D, where

$$\begin{aligned} D = ((N, \mathbb{G}, \mathbb{G}_T, e), g, g_2, g_3, g^d, g^{da^q}, g^{cd}, g^a, g^{a^2}, \ldots, g^{a^q}, \\ g^{a^i/b_j} \quad \forall i \in [2q] \setminus \{q+1\}, j \in [q], \\ g^{cdb_j} \quad \forall j \in [q], \\ g^{cda^i b_{j'}/b_j} \quad \forall i \in [q], \ j, j' \in [q] \ s.t. \ j \neq j' \ ), \end{aligned}$$

in which $(N = p_1p_2p_3, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}$, $g \xleftarrow{R} \mathbb{G}_{p_1}$, $g_2 \xleftarrow{R} \mathbb{G}_{p_2}, g_3 \xleftarrow{R} \mathbb{G}_{p_3}, a, c, d, b_1, \ldots, b_q \xleftarrow{R} \mathbb{Z}_N$, and T is either equal to ca^{q+1} or a random element of $\mathbb{G}_{p_1}$. $\mathcal{B}$'s goal is to determine whether $T = g^{ca^{q+1}}$ or a random element from $\mathbb{G}_{p_1}$.

Init: The adversary gives $\mathcal{B}$ the challenge LSSS matrix (A^*, ρ^*), where A^* is an $l \times n$ matrix, and $l, n \leq q$.

Setup: For any $x \in \mathcal{U}$, let $J_x = \{k | \rho^*(k) = x\}$. $\mathcal{B}$ randomly chooses

$$\begin{aligned} &\{\alpha_i \in \mathbb{Z}_N\}_{i \in [m]}, \ \{r_i, \ z'_i \in \mathbb{Z}_N\}_{i \in [m] \setminus \{\bar{i}\}}, \ \ r'_{\bar{i}}, \ z_{\bar{i}}, \\ &\{c'_j \in \mathbb{Z}_N\}_{j \in [m]}, \\ &\{a_x \in \mathbb{Z}_N\}_{x \in \mathcal{U} \ s.t. \ J_x = \emptyset}, \quad \{a'_x \in \mathbb{Z}_N\}_{x \in \mathcal{U} \ s.t. \ J_x \neq \emptyset}, \end{aligned}$$

and $\theta \in \mathbb{Z}_N$. $\mathcal{B}$ gives $\mathcal{A}$ the public parameter PP:

$$\begin{aligned} \Big(&g, \ f = g^a, \ h = g^\theta, \ \{E_i = e(g,g)^{\alpha_i}\}_{i \in [m]}, \\ &\{G_i = g^{r_i}, \ Z_i = (g^a)^{z'_i}\}_{i \in [m] \setminus \{\bar{i}\}}, \ G_{\bar{i}} = (g^{a^q})^{r'_{\bar{i}}}, \ Z_{\bar{i}} = g^{z_{\bar{i}}}, \\ &\{H_j = (g^d)^{c'_j}\}_{j \in [m] \setminus \{\bar{j}\}}, \ H_{\bar{j}} = (g^a)^{c'_{\bar{j}}}, \\ &\{U_x = g^{a_x}\}_{x \in \mathcal{U} \ s.t. \ J_x = \emptyset}, \\ &\{U_x = g^{a'_x} \prod_{k \in J_x} \prod_{t \in [n]} (g^{a^t/b_k})^{A^*_{k,t}}\}_{x \in \mathcal{U} \ s.t. \ J_x \neq \emptyset}\Big). \end{aligned}$$

Note that $\mathcal{B}$ implicitly chooses $r_{\bar{i}}, \{c_j \in \mathbb{Z}_N\}_{j \in [m]}$, $\{z_i \in \mathbb{Z}_N\}_{i \in [m] \setminus \{\bar{i}\}}$ and $\{a_x \in \mathbb{Z}_N\}_{x \in \mathcal{U} \ s.t. \ J_x \neq \emptyset}$ such that

$$\begin{aligned} &a^q r'_{\bar{i}} \equiv r_{\bar{i}} \bmod p_1, \\ &dc'_j \equiv c_j \bmod p_1 \quad \forall j \in [m] \setminus \{\bar{j}\}, \quad ac'_{\bar{j}} \equiv c_{\bar{j}} \bmod p_1, \\ &az'_i \equiv z_i \bmod p_1 \quad \forall i \in [m] \setminus \{\bar{i}\}, \\ &a'_x + \sum_{k \in J_x} \sum_{t \in [n]} (a^t/b_k) A^*_{k,t} \equiv a_x \bmod p_1 \quad \forall x \in \mathcal{U} \ s.t. \ J_x \neq \emptyset. \end{aligned}$$

Key Query: To respond to $\mathcal{A}$'s query for $((i,j), S_{(i,j)})$,

- *if* $(i,j) \neq (\bar{i}, \bar{j})$*:* $\mathcal{B}$ randomly chooses $\sigma_{i,j}$, $\delta_{i,j} \in \mathbb{Z}_N$ and $R, R', R'', R''', R_x (x \in S_{(i,j)}) \in \mathbb{G}_{p_3}$, then creates the private key $\mathsf{SK}_{(i,j), S_{(i,j)}} = \big((i,j), S_{(i,j)}, K_{i,j}, \ K'_{i,j}, \ K''_{i,j}, \ K'''_{i,j}, \ \{K_{i,j,x}\}_{x \in S_{(i,j)}}\big)$ as

$$K_{i,j} = \begin{cases} g^{\alpha_i} (g^d)^{r_i c'_j} f^{\sigma_{i,j}} h^{\delta_{i,j}} R, : & i \neq \bar{i}, j \neq \bar{j} \\ g^{\alpha_i} (g^{da^q})^{r'_{\bar{i}} c'_j} f^{\sigma_{i,j}} h^{\delta_{i,j}} R, : & i = \bar{i}, j \neq \bar{j} \\ g^{\alpha_i} (g^a)^{r_i c'_{\bar{j}}} f^{\sigma_{i,j}} h^{\delta_{i,j}} R, : & i \neq \bar{i}, j = \bar{j} \end{cases}$$

$$K'_{i,j} = g^{\sigma_{i,j}} R', \ K''_{i,j} = g^{\delta_{i,j}} R'', \ K'''_{i,j} = Z_i^{\sigma_{i,j}} R''',$$

$$K_{i,j,x} = U_x^{\sigma_{i,j}} R_x \quad \forall x \in S_{(i,j)}.$$

- *if* $(i,j) = (\bar{i}, \bar{j})$*:* it means that $\mathcal{A}$ is querying a private key with the challenge index $(\bar{i}, \bar{j})$, and $S_{(i,j)}$ does not satisfy (A^*, ρ^*). $\mathcal{B}$ first computes a vector $\vec{\bar{u}} = (\bar{u}_1, \ldots, \bar{u}_n) \in \mathbb{Z}_N^n$ that has the first entry equal to $-r'_{\bar{i}}c'_{\bar{j}}$ (i.e. $\bar{u}_1 = -r'_{\bar{i}}c'_{\bar{j}}$) and is orthogonal to all of the rows A^*_k of A^* such that $\rho^*(k) \in S_{(i,j)}$ (i.e. $A^*_k \cdot \vec{\bar{u}} = 0 \ \forall k \in [l]$ $s.t.$ $\rho^*(k) \in S_{(i,j)}$). Such a vector must exist as $S_{(i,j)}$ does not satisfy (A^*, ρ^*), and is efficiently computable. Then $\mathcal{B}$ chooses a random $\sigma'_{\bar{i},\bar{j}} \in \mathbb{Z}_N$ and sets the value of $\sigma_{\bar{i},\bar{j}}$ by implicitly setting

$$\sigma'_{\bar{i},\bar{j}} + (\bar{u}_1 a^q + \bar{u}_2 a^{q-1} + \cdots + \bar{u}_n a^{q-n+1}) \equiv \sigma_{\bar{i},\bar{j}} \bmod p_1.$$

In addition $\mathcal{B}$ randomly chooses $\delta_{\bar{i},\bar{j}} \in \mathbb{Z}_N$ and $R, R', R'', R''', R_x (x \in S_{(i,j)}) \in \mathbb{G}_{p_3}$. $\mathcal{B}$ creates the private key $\mathsf{SK}_{(\bar{i},\bar{j}),S_{(\bar{i},\bar{j})}} = \big((\bar{i},\bar{j}), S_{(\bar{i},\bar{j})}, \ K_{\bar{i},\bar{j}}, \ K'_{\bar{i},\bar{j}}, \ K''_{\bar{i},\bar{j}}, \ K'''_{\bar{i},\bar{j}}, \ \{K_{\bar{i},\bar{j},x}\}_{x \in S_{(\bar{i},\bar{j})}}\big)$ as follows:

$$K_{\bar{i},\bar{j}} = g^{\alpha_{\bar{i}}} f^{\sigma'_{\bar{i},\bar{j}}} \Big(\prod_{2 \le t \le n} (g^{a^{q-t+2}})^{\bar{u}_t} \Big) h^{\delta_{\bar{i},\bar{j}}} R,$$

$$K'_{\bar{i},\bar{j}} = g^{\sigma'_{\bar{i},\bar{j}}} \Big(\prod_{t \in [n]} (g^{a^{q-t+1}})^{\bar{u}_t} \Big) R',$$

$$K''_{\bar{i},\bar{j}} = g^{\delta_{\bar{i},\bar{j}}} R'', \quad K'''_{\bar{i},\bar{j}} = g^{\sigma'_{\bar{i},\bar{j}} z_{\bar{i}}} \Big(\prod_{t \in [n]} (g^{a^{q-t+1}})^{\bar{u}_t} \Big)^{z_{\bar{i}}} R'''.$$

For $x \in S_{(i,j)}$ $s.t.$ $J_x = \emptyset$, we have

$$\begin{aligned} K_{\bar{i},\bar{j},x} &= U_x^{\sigma_{\bar{i},\bar{j}}} R_x = U_x^{\sigma'_{\bar{i},\bar{j}}} (g^{a_x})^{\sum_{t \in [n]} \bar{u}_t a^{q-t+1}} R_x \\ &= U_x^{\sigma'_{\bar{i},\bar{j}}} \Big(\prod_{t \in [n]} (g^{a^{q-t+1}})^{\bar{u}_t} \Big)^{a_x} R_x, \end{aligned}$$

i.e., $\mathcal{B}$ can compute $K_{\bar{i},\bar{j},x}$.
For $x \in S_{(i,j)}$ $s.t.$ $J_x \neq \emptyset$, we have

$$\begin{aligned} K_{\bar{i},\bar{j},x} &= U_x^{\sigma_{\bar{i},\bar{j}}} R_x \\ &= U_x^{\sigma'_{\bar{i},\bar{j}}} \Big(g^{a'_x} \prod_{k \in J_x} \prod_{t \in [n]} (g^{a^t/b_k})^{A^*_{k,t}} \Big)^{\sum_{t' \in [n]} \bar{u}_{t'} a^{q-t'+1}} R_x \\ &= U_x^{\sigma'_{\bar{i},\bar{j}}} \Big(\prod_{t' \in [n]} (g^{a^{q-t'+1}})^{\bar{u}_{t'}} \Big)^{a'_x} \\ &\quad \cdot \Big(\prod_{k \in J_x} \prod_{t \in [n]} \prod_{t' \in [n]} (g^{a^{q-t'+t+1}/b_k})^{A^*_{k,t} \bar{u}_{t'}} \Big) R_x \\ &= U_x^{\sigma'_{\bar{i},\bar{j}}} \Big(\prod_{t' \in [n]} (g^{a^{q-t'+1}})^{\bar{u}_{t'}} \Big)^{a'_x} \\ &\quad \cdot \Big(\prod_{k \in J_x} \prod_{t \in [n]} \prod_{t' \in [n] \setminus \{t\}} (g^{a^{q-t'+t+1}/b_k})^{A^*_{k,t} \bar{u}_{t'}} \Big) \\ &\quad \cdot \Big(\prod_{k \in J_x} \prod_{t \in [n]} (g^{a^{q+1}/b_k})^{A^*_{k,t} \bar{u}_t} \Big) R_x \\ &= U_x^{\sigma'_{\bar{i},\bar{j}}} \Big(\prod_{t' \in [n]} (g^{a^{q-t'+1}})^{\bar{u}_{t'}} \Big)^{a'_x} \\ &\quad \cdot \Big(\prod_{k \in J_x} \prod_{t \in [n]} \prod_{t' \in [n] \setminus \{t\}} (g^{a^{q-t'+t+1}/b_k})^{A^*_{k,t} \bar{u}_{t'}} \Big) \\ &\quad \cdot \Big(\prod_{k \in J_x} (g^{a^{q+1}/b_k})^{\sum_{t \in [n]} A^*_{k,t} \bar{u}_t} \Big) R_x, \\ &= U_x^{\sigma'_{\bar{i},\bar{j}}} \Big(\prod_{t' \in [n]} (g^{a^{q-t'+1}})^{\bar{u}_{t'}} \Big)^{a'_x} \\ &\quad \cdot \Big(\prod_{k \in J_x} \prod_{t \in [n]} \prod_{t' \in [n] \setminus \{t\}} (g^{a^{q-t'+t+1}/b_k})^{A^*_{k,t} \bar{u}_{t'}} \Big) R_x, \\ &\quad \text{(since for } \ k \in J_x \text{ where } x \in S_{(i,j)}, \text{ we have} \\ &\quad \sum_{t \in [n]} A^*_{k,t} \bar{u}_t = A^*_k \cdot \vec{\bar{u}} = 0) \end{aligned}$$

i.e., $\mathcal{B}$ can also compute $K_{\bar{i},\bar{j},x}$.

Challenge: $\mathcal{A}$ submits a message M. $\mathcal{B}$ randomly chooses

$$\begin{aligned} &\tau', \quad s_1, \ldots, s_{\bar{i}-1}, s'_{\bar{i}}, s_{\bar{i}+1}, \ldots, s_m \in \mathbb{Z}_N, \\ &t'_1, \ldots, t'_{\bar{i}-1}, t_{\bar{i}}, t'_{\bar{i}+1}, \ldots, t'_m \in \mathbb{Z}_N, \\ &\vec{w}_1, \ldots, \vec{w}_{\bar{j}-1}, \vec{w}'_{\bar{j}}, \ldots, \vec{w}'_m \in \mathbb{Z}_N^3, \\ &\xi'_k \in \mathbb{Z}_N \quad \forall k \in [l], \\ &\pi' \in \mathbb{Z}_N, \quad \vec{u}' = (0, u'_2, \ldots, u'_n) \ \in \mathbb{Z}_N^n. \end{aligned}$$

It also randomly chooses $r_x, r_y, r_z \in \mathbb{Z}_N$, and sets $\vec{\chi}_1 = (r_x, 0, r_z), \vec{\chi}_2 = (0, r_y, r_z), \vec{\chi}_3 = \vec{\chi}_1 \times \vec{\chi}_2 = (-r_y r_z, -r_x r_z, r_x r_y)$. Then it randomly chooses

$$\begin{aligned} &\vec{v}_i \in \mathbb{Z}_N^3 \quad \forall i \in \{1, \ldots, \bar{i}-1\}, \\ &\vec{v}^p_{\bar{i}} \in span\{\vec{\chi}_1, \vec{\chi}_2\}, \ \vec{v}^q_{\bar{i}} \in span\{\vec{\chi}_3\}, \\ &\vec{v}_i \in span\{\vec{\chi}_1, \vec{\chi}_2\} \quad \forall i \in \{\bar{i}+1, \ldots, m\}. \end{aligned}$$

$\mathcal{B}$ randomly chooses $\vec{v}^p_c \in span\{\vec{\chi}_1, \vec{\chi}_2\}$, $\vec{v}^q_c \in span\{\vec{\chi}_3\}$, where $\vec{v}^q_c = \nu_c \vec{\chi}_3$ for randomly chosen $\nu_c \in \mathbb{Z}_N$. $\mathcal{B}$ sets the values of $\kappa, \tau, \ s_{\bar{i}}, \ t_i (i \in [m] \setminus \{\bar{i}\}) \in \mathbb{Z}_N, \vec{v}_{\bar{i}}, \ \vec{v}_c, \ \vec{w}_j (j \in \{\bar{j}, \ldots, m\}) \in \mathbb{Z}_N^3$, $\pi \in \mathbb{Z}_N$, $\vec{u} \in \mathbb{Z}_N^n$, and $\{\xi_k \in \mathbb{Z}_N\}_{k \in [l]}$ by implicitly setting

$$\begin{aligned} &a^q \equiv \kappa \bmod p_1, \quad c a^q \tau' \equiv \tau \bmod p_1, \quad s'_{\bar{i}}/a^q \equiv s_{\bar{i}} \bmod p_1, \\ &\forall i \in \{1, \ldots, \bar{i}-1\}: \quad t'_i + c d \tau' s'_{\bar{i}} (\vec{v}^q_{\bar{i}} \cdot \vec{v}^q_c)/z'_i \equiv t_i \bmod p_1, \\ &\forall i \in \{\bar{i}+1, \ldots, m\}: \\ &\quad t'_i - a^q \tau' s_i (\vec{v}_i \cdot \vec{v}^p_c)/z'_i + c d \tau' s'_{\bar{i}} (\vec{v}^q_{\bar{i}} \cdot \vec{v}^q_c)/z'_i \equiv t_i \bmod p_1, \\ &\qquad \vec{v}_{\bar{i}} = \vec{v}^p_{\bar{i}} + d \vec{v}^q_{\bar{i}}, \\ &\qquad \vec{v}_c = c^{-1} \vec{v}^p_c + \vec{v}^q_c, \\ &\qquad \vec{w}'_{\bar{j}} - a c'_{\bar{j}} \tau' \vec{v}^p_c \equiv \vec{w}_{\bar{j}} \bmod p_1, \\ &\forall j \in \{\bar{j}+1, \ldots, m\}: \quad \vec{w}'_j - c d c'_j \tau' \vec{v}^q_c \equiv \vec{w}_j \bmod p_1, \\ &\qquad \pi' - c d \tau' s'_{\bar{i}} (\vec{v}^q_{\bar{i}} \cdot \vec{v}^q_c) \equiv \pi \bmod p_1, \\ &\qquad \vec{u} = \pi (1, a, a^2, \ldots, a^{n-1}) + \vec{u}', \\ &\forall k \in [l]: \quad \xi'_k - c d b_k \tau' s'_{\bar{i}} (\vec{v}^q_{\bar{i}} \cdot \vec{v}^q_c) \equiv \xi_k \bmod p_1. \end{aligned}$$

Note that $\vec{v}_{\bar{i}} = \vec{v}^p_{\bar{i}} + d\vec{v}^q_{\bar{i}}$ and $\vec{v}_c = c^{-1}\vec{v}^p_c + \vec{v}^q_c$ are vectors chosen randomly from $\mathbb{Z}_N^3$ as required: as $\vec{\chi}_3$ is orthogonal to $span\{\vec{\chi}_1, \vec{\chi}_2\}$ and $\mathbb{Z}_N^3 = span\{\vec{\chi}_1, \vec{\chi}_2, \vec{\chi}_3\}$, we can choose a random $\vec{v} \in \mathbb{Z}_N^3$ by choosing random $\vec{v}^p \in span\{\vec{\chi}_1, \vec{\chi}_2\}$ and $\vec{v}^q \in span\{\vec{\chi}_3\}$ and setting $\vec{v} = \vec{v}^p + \vec{v}^q$. Also note that we have $(\vec{v}_{\bar{i}} \cdot \vec{v}_c) = c^{-1}(\vec{v}^p_{\bar{i}} \cdot \vec{v}^p_c) + d(\vec{v}^q_{\bar{i}} \cdot \vec{v}^q_c)$ since $\vec{v}^p_{\bar{i}}, \vec{v}^p_c \in span\{\vec{\chi}_1, \vec{\chi}_2\}$, $\vec{v}^q_{\bar{i}}, \vec{v}^q_c \in span\{\vec{\chi}_3\}$.

$\mathcal{B}$ creates the ciphertext $\langle (A, \rho), \ (\vec{R}_i, \vec{R}'_i, Q_i, Q'_i, Q''_i, Q'''_i, T_i)_{i=1}^m, \ (\vec{C}_j, \vec{C}'_j)_{j=1}^m, \ (P_k, P'_k)_{k=1}^l \rangle$ as follows:

1) For each row $i \in [m]$:
   - if $i < \bar{i}$: it randomly chooses $\hat{s}_i \in \mathbb{Z}_N$, then sets

$$\begin{aligned}
\vec{R}_i &= g^{\vec{v}_i},\ \ \vec{R}'_i = (g^{a^q})^{\vec{v}_i},\\
Q_i &= g^{s_i},\ \ Q'_i = f^{s_i} Z_i^{t'_i} f^{\pi'},\ \ Q''_i = h^{s_i},\\
Q'''_i &= g^{t'_i}(g^{cd})^{\tau' s'_{\bar{i}}(\vec{v}_{\bar{i}}^q \cdot \vec{v}_c^q)/z'_i},\quad T_i = E_i^{\hat{s}_i}.
\end{aligned}$$

   - if $i = \bar{i}$: it sets

$$\begin{aligned}
\vec{R}_i &= g^{r'_{\bar{i}} s'_{\bar{i}} \vec{v}_{\bar{i}}^p} \cdot (g^d)^{r'_{\bar{i}} s'_{\bar{i}} \vec{v}_{\bar{i}}^q},\\
\vec{R}'_i &= (g^{a^q})^{r'_{\bar{i}} s'_{\bar{i}} \vec{v}_{\bar{i}}^p} \cdot (g^{da^q})^{r'_{\bar{i}} s'_{\bar{i}} \vec{v}_{\bar{i}}^q},\\
Q_i &= g^{\tau' s'_{\bar{i}}(\vec{v}_{\bar{i}}^p \cdot \vec{v}_c^p)}(g^{cd})^{\tau' s'_{\bar{i}}(\vec{v}_{\bar{i}}^q \cdot \vec{v}_c^q)},\\
Q'_i &= f^{\tau' s'_{\bar{i}}(\vec{v}_{\bar{i}}^p \cdot \vec{v}_c^p)} Z_i^{t_{\bar{i}}} f^{\pi'},\ \ Q''_i = Q_i^{\theta},\ \ Q'''_i = g^{t_{\bar{i}}},\\
T_i &= M \cdot e(g^{\alpha_i}, Q_i).
\end{aligned}$$

   - if $i > \bar{i}$: it sets

$$\begin{aligned}
\vec{R}_i &= g^{r_i s_i \vec{v}_i},\quad \vec{R}'_i = (g^{a^q})^{r_i s_i \vec{v}_i},\\
Q_i &= (g^{a^q})^{\tau' s_i (\vec{v}_i \cdot \vec{v}_c^p)},\ \ Q'_i = Z_i^{t'_i} f^{\pi'},\ \ Q''_i = Q_i^{\theta},\\
Q'''_i &= g^{t'_i}\big((g^{a^q})^{-s_i(\vec{v}_i \cdot \vec{v}_c^p)}(g^{cd})^{s'_{\bar{i}}(\vec{v}_{\bar{i}}^q \cdot \vec{v}_c^q)}\big)^{\tau'/z'_i},\\
T_i &= M \cdot e(g^{\alpha_i}, Q_i).
\end{aligned}$$

2) For each $j \in [m]$:
   - if $j < \bar{j}$: it randomly chooses $\mu'_j \in \mathbb{Z}_N$ and implicitly sets the value of μ_j such that $(\mu'_j/(cda^q) - 1)\nu_c \equiv \mu_j \bmod N$, then sets

$$\vec{C}_j = (g^{da^q})^{c'_j \tau' \vec{v}_c^p} \cdot g^{c'_j \tau' \mu'_j \vec{v}_c^q} \cdot (g^{a^q})^{\vec{w}_j},\quad \vec{C}'_j = g^{\vec{w}_j}.$$

   - if $j = \bar{j}$: $\vec{C}_j = T^{c'_{\bar{j}} \tau' \vec{v}_c^q} \cdot (g^{a^q})^{\vec{w}'_j}, \vec{C}'_j = g^{\vec{w}'_{\bar{j}}} \cdot (g^a)^{-c'_{\bar{j}} \tau' \vec{v}_c^p}$.
   - if $j > \bar{j}$: $\vec{C}_j = (g^{da^q})^{c'_j \tau' \vec{v}_c^p} \cdot (g^{a^q})^{\vec{w}'_j},\quad \vec{C}'_j = g^{\vec{w}'_j} \cdot (g^{cd})^{-c'_j \tau' \vec{v}_c^q}$.

3) For each $k \in [l]$: we have

$$\begin{aligned}
P_k &= f^{A_k^* \cdot \vec{u}} U_{\rho^*(k)}^{-\xi_k}\\
&= \big(f^{A_k^* \cdot (1,a,\ldots,a^{n-1})}\big)^{\pi} f^{A_k^* \cdot \vec{u}'}\\
&\quad \cdot \Big(g^{a'_{\rho^*(k)}} \prod_{k' \in J_{\rho^*(k)}} \prod_{t \in [n]} (g^{a^t/b_{k'}})^{A^*_{k',t}}\Big)^{-\xi_k}\\
&= \Big(\prod_{t \in [n]} (g^{a^t})^{A^*_{k,t}}\Big)^{\pi' - cd\tau' s'_{\bar{i}}(\vec{v}_{\bar{i}}^q \cdot \vec{v}_c^q)} f^{A_k^* \cdot \vec{u}'}\\
&\quad \cdot \Big(g^{a'_{\rho^*(k)}} \prod_{k' \in J_{\rho^*(k)}} \prod_{t \in [n]} (g^{a^t/b_{k'}})^{A^*_{k',t}}\Big)^{-\xi'_k}\\
&\quad \cdot \Big(g^{a'_{\rho^*(k)}} \prod_{k' \in J_{\rho^*(k)}} \prod_{t \in [n]} (g^{a^t/b_{k'}})^{A^*_{k',t}}\Big)^{cdb_k \tau' s'_{\bar{i}}(\vec{v}_{\bar{i}}^q \cdot \vec{v}_c^q)}\\
&= \Big(\prod_{t \in [n]} (g^{a^t})^{A^*_{k,t}}\Big)^{\pi'} \Big(\prod_{t \in [n]} (g^{cda^t})^{A^*_{k,t}}\Big)^{-\tau' s'_{\bar{i}}(\vec{v}_{\bar{i}}^q \cdot \vec{v}_c^q)}\\
&\quad \cdot f^{A_k^* \cdot \vec{u}'} g^{-a'_{\rho^*(k)} \xi'_k} (g^{cdb_k})^{a'_{\rho^*(k)} \tau' s'_{\bar{i}}(\vec{v}_{\bar{i}}^q \cdot \vec{v}_c^q)}\\
&\quad \cdot \Big(\prod_{k' \in J_{\rho^*(k)}} \prod_{t \in [n]} (g^{a^t/b_{k'}})^{A^*_{k',t}}\Big)^{-\xi'_k}\\
&\quad \cdot \Big(\prod_{k' \in J_{\rho^*(k)}} \prod_{t \in [n]} (g^{cda^t b_k/b_{k'}})^{A^*_{k',t}}\Big)^{\tau' s'_{\bar{i}}(\vec{v}_{\bar{i}}^q \cdot \vec{v}_c^q)}\\
&= \Big(\prod_{t \in [n]} (g^{a^t})^{A^*_{k,t}}\Big)^{\pi'} \Big(\prod_{t \in [n]} (g^{cda^t})^{A^*_{k,t}}\Big)^{-\tau' s'_{\bar{i}}(\vec{v}_{\bar{i}}^q \cdot \vec{v}_c^q)}\\
&\quad \cdot f^{A_k^* \cdot \vec{u}'} g^{-a'_{\rho^*(k)} \xi'_k} (g^{cdb_k})^{a'_{\rho^*(k)} \tau' s'_{\bar{i}}(\vec{v}_{\bar{i}}^q \cdot \vec{v}_c^q)}\\
&\quad \cdot \Big(\prod_{k' \in J_{\rho^*(k)}} \prod_{t \in [n]} (g^{a^t/b_{k'}})^{A^*_{k',t}}\Big)^{-\xi'_k}\\
&\quad \cdot \Big(\prod_{k' \in J_{\rho^*(k)} \backslash \{k\}} \prod_{t \in [n]} (g^{cda^t b_k/b_{k'}})^{A^*_{k',t}}\Big)^{\tau' s'_{\bar{i}}(\vec{v}_{\bar{i}}^q \cdot \vec{v}_c^q)}\\
&\quad \cdot \Big(\prod_{t \in [n]} (g^{cda^t b_k/b_k})^{A^*_{k,t}}\Big)^{\tau' s'_{\bar{i}}(\vec{v}_{\bar{i}}^q \cdot \vec{v}_c^q)}\\
&= \Big(\prod_{t \in [n]} (g^{a^t})^{A^*_{k,t}}\Big)^{\pi'} f^{A_k^* \cdot \vec{u}'} g^{-a'_{\rho^*(k)} \xi'_k}\\
&\quad \cdot (g^{cdb_k})^{a'_{\rho^*(k)} \tau' s'_{\bar{i}}(\vec{v}_{\bar{i}}^q \cdot \vec{v}_c^q)}\\
&\quad \cdot \Big(\prod_{k' \in J_{\rho^*(k)}} \prod_{t \in [n]} (g^{a^t/b_{k'}})^{A^*_{k',t}}\Big)^{-\xi'_k}\\
&\quad \cdot \Big(\prod_{k' \in J_{\rho^*(k)} \backslash \{k\}} \prod_{t \in [n]} (g^{cda^t b_k/b_{k'}})^{A^*_{k',t}}\Big)^{\tau' s'_{\bar{i}}(\vec{v}_{\bar{i}}^q \cdot \vec{v}_c^q)},
\end{aligned}$$

$$P'_k = g^{\xi_k} = g^{\xi'_k}(g^{cdb_k})^{-\tau' s'_{\bar{i}}(\vec{v}_{\bar{i}}^q \cdot \vec{v}_c^q)},$$

i.e., $\mathcal{B}$ can compute the values of P_k and P'_k.

If $T = g^{ca^{q+1}}$, the ciphertext is a well-formed encryption to the index $(\bar{i}, \bar{j})$. If T is randomly chosen, say $T = g^{ca^{q+1}+r}$ for some random $r \in \mathbb{Z}_N$, the ciphertext is a well-formed encryption to the index $(\bar{i}, \bar{j}+1)$ with implicitly setting $\mu_{\bar{j}}$ such that $(r\nu_c)/(ca^{q+1}) \equiv \mu_{\bar{j}} \bmod p_1$.

Guess: $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ to $\mathcal{B}$, then $\mathcal{B}$ outputs this b' to the challenger. The distributions of the public parameters, private keys and challenge ciphertext are the same as that in the real scheme. $\mathcal{B}$'s advantage in the modified source group q-parallel BDHE game will be exactly equal to $\mathcal{A}$'s advantage in the selective index-hiding game. □

Appendix B
Proof of the Modified Source Group q-Parallel BDHE Assumption

When compared with the Source Group q-Parallel BDHE Assumption [12], the adversary in our Modified Source Group q-Parallel BDHE Assumption is given an additional term fc^q. In this section, we give a lower bound to the complexity of our modified source group parallel BDHE assumption for the prime order version, though the proof for the composite-order version in a subgroup is analogous. The proof is similar to that of the original one [12], which is given in [19, Appendix B] in the generic group model. In the generic group model [20], an adversary does not have direct access to the group. It must interact with an oracle to perform the group operation and obtain "handles" for new elements. Also, it can only use handles previously received from the oracle. We consider an experiment where an adversary is given handles for the group elements given out in the assumption as well as a handle for the challenge term T_β (here, β is a uniformly random bit). The adversary may interact with the oracle to perform group operations and pairings, and gets handles in return as the results from these operations. Finally, the adversary guesses the bit β. The difference between the adversary's success probability and one half is defined as its advantage. We refer readers to [21], [22] for other examples of using the generic group model for justifying assumptions in bilinear groups. We denote $c, d, f, b_1, \ldots, b_q$ as variables over $\mathbb{Z}_p$, and define $\mathcal{M}$ as the following set of rational functions over these variables:

$$\begin{aligned}\mathcal{M} := \{1, f, df, fc^q, c^1, \ldots, c^q,\\ c^i/b_j \quad \forall i \in [2q] \setminus \{q+1\}, j \in [q],\\ dfb_j \quad \forall j \in [q],\\ dfc^i b_{j'}/b_j \quad \forall i \in [q], j, j' \in [q]\ s.t.\ j \neq j'\}.\end{aligned}$$

These are the exponents of the group elements given in our modified source group q-parallel BDHE assumption. Let $E(\mathcal{M})$ be the set of all pairwise products of functions in $\mathcal{M}$. It represents the exponents of elements in $\mathbb{G}_T$ that can be obtained by pairing elements with exponents in $\mathcal{M}$. We say a function T is *dependent* on a set of functions $\mathcal{S} = \{S_1, \ldots, S_k\}$ if there exist constants $r_1, \ldots, r_k \in \mathbb{Z}_p$ such that $T = r_1 S_1 + \cdots + r_k S_k$. This is an equality of functions over $\mathbb{Z}_p$, and hence hold for *all* settings of the variables. If no such constants exist, we say that T is *independent* of $\mathcal{S}$.

Lemma 3: For each function $M \in \mathcal{M} \cup \{dc^{q+1}\}$, the product $M \cdot dc^{q+1}$ is independent of $E(\mathcal{M}) \cup dc^{q+1}(\mathcal{M} \setminus M)$. (Here, $dc^{q+1}(\mathcal{M} \setminus M)$ denotes the set formed by removing M from $\mathcal{M}$ and then multiplying all remaining elements by dc^{q+1}.)

Proof: As every element in $\mathcal{M} \cup \{dc^{q+1}\}$ and $E(\mathcal{M}) \cup dc^{q+1}(\mathcal{M} \setminus M)$ is a ratio of monomials, the only way that $M(dc^{q+1})$ can be dependent on $E(\mathcal{M}) \cup dc^{q+1}(\mathcal{M} \setminus M)$ is if it is *contained* in $E(\mathcal{M}) \cup dc^{q+1}(\mathcal{M} \setminus M)$. First, d^2c^{2q+2} is not in $E(\mathcal{M}) \cup dc^{q+1}\mathcal{M}$, and for any $M \in \mathcal{M}$, $dc^{q+1}M \notin dc^{q+1}(\mathcal{M} \setminus M)$. Thus it suffices to show that for any M, $dc^{q+1}M \notin E(\mathcal{M})$. In other words, we show that $E(\mathcal{M})$ does not intersect with the set $dc^{q+1}\mathcal{M}$, which is formed by multiplying each element of $\mathcal{M}$ by dc^{q+1}. To see this, we examine the set $dc^{q+1}\mathcal{M}$. By definition, we have that

$$\begin{aligned}dc^{q+1}\mathcal{M} = \{dc^{q+1}, dfc^{q+1}, d^2fc^{q+1}, dfc^{2q+1},\\ dc^{q+2}, \ldots, dc^{2q+1},\\ dc^{q+1+i}/b_j \quad \forall i \in [2q] \setminus \{q+1\}, j \in [q],\\ d^2fb_jc^{q+1} \quad \forall j \in [q],\\ d^2fc^{q+1+i}b_{j'}/b_j \quad \forall i \in [q], j, j' \in [q]\ s.t.\ j \neq j'\}.\end{aligned}$$

We now check if any of these are in $E(\mathcal{M})$, which is the set of pairwise products of things in $\mathcal{M}$. In $\mathcal{M}$, every occurrence of d is accompanied by f, and f^{-1} never appears. Hence $E(\mathcal{M})$ does not contain any element which has a higher power of d than f. This rules out all the elements in $dc^{q+1}\mathcal{M}$ above but dfc^{q+1} and dfc^{2q+1}. To rule out dfc^{q+1}, we consider all the possible ways it might be formed as a product of two elements of $\mathcal{M}$. As f is in the term, one of the two factors in $\mathcal{M}$ must be a term containing f. Note that f, df, or fc^q cannot be one of the factors as $dc^{q+1}, c^{q+1}, dc \notin \mathcal{M}$. Also, an element of the form dfb_j cannot be one of the two factors as $c^{q+1}/b_j \notin \mathcal{M}$, and an element of the form $dfc^ib_{j'}/b_j$ $(s.t.\ j' \neq j)$ cannot be one of the factors as $c^{q+1-i}b_j/b_{j'} \notin \mathcal{M}$. Hence we can dismiss all the possible ways, and conclude that $dfc^{q+1} \notin E(\mathcal{M})$. To rule out dfc^{2q+1}, we consider all the possible ways it might be formed as a product of two elements of $\mathcal{M}$. Since f is in the term, one of the two factors in $\mathcal{M}$ must be a term containing f. Similarly, f, df, or fc^q cannot be one of the factors as $dc^{2q+1}, c^{2q+1}, dc^{q+1} \notin \mathcal{M}$. An element of the form dfb_j cannot be one of the factors as $c^{2q+1}/b_j \notin \mathcal{M}$. An element of the form $dfc^ib_{j'}/b_j$ $(s.t.\ j' \neq j)$ cannot be either as $c^{2q+1-i}b_j/b_{j'} \notin \mathcal{M}$. Hence we can dismiss all the possible ways, and conclude that $dfc^{2q+1} \notin E(\mathcal{M})$. □

We now proceed similarly to the proof strategy in [21], [22] to establish the following theorem:

Theorem 6: For any adversary $\mathcal{A}$ that makes Q queries to the oracles computing the group operations in $\mathbb{G}, \mathbb{G}_T$ and the bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, the advantages of $\mathcal{A}$ against the modified source group q-parallel BDHE assumption in the generic group model is at most $O(\frac{Q^2q}{p})$.

Proof: The proof of this theorem is identical to that of Theorem 22 in [19]. □

References

[1] 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.

[2] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*. Berlin, Germany: Springer-Verlag, 1984, pp. 47–53.

[3] D. Boneh and M. K. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology*. Berlin, Germany: Springer-Verlag, 2001, pp. 213–229.

[4] M. Naor and B. Pinkas, "Efficient trace and revoke schemes," in *Financial Cryptography*. Berlin, Germany: Springer-Verlag, 2000, pp. 1–20.

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

[6] L. Cheung and C. C. Newport, "Provably secure ciphertext policy ABE," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, 2007, pp. 456–465.

[7] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in *Proc. 35th ICALP*, 2008, pp. 579–591.

[8] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Public Key Cryptography*. Berlin, Germany: Springer-Verlag, 2011, pp. 53–70.

[9] 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 *Advances in Cryptology*. Berlin, Germany: Springer-Verlag, 2010, pp. 62–91.

[10] T. Okamoto and K. Takashima, "Fully secure functional encryption with general relations from the decisional linear assumption," in *Advances in Cryptology*. Berlin, Germany: Springer-Verlag, 2010, pp. 191–208.

[11] J. Herranz, F. Laguillaumie, and C. Ràfols, "Constant size ciphertexts in threshold attribute-based encryption," in *Public Key Cryptography*. Berlin, Germany: Springer-Verlag, 2010, pp. 19–34.

[12] 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-Verlag, 2012, pp. 180–198.

[13] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 463–474.

[14] 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 Security*, vol. 8, no. 1, pp. 76–88, Jan. 2013.

[15] 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.

[16] 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-Verlag, 2006, pp. 573–592.

[17] D. Boneh and B. Waters, "A fully collusion resistant broadcast, trace, and revoke system," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, 2006, pp. 211–220.

[18] S. Garg, A. Kumarasubramanian, A. Sahai, and B. Waters, "Building efficient fully collusion-resilient traitor tracing and revocation schemes," in *Proc. 17th ACM Conf. Comput. Commun. Secur.*, 2010, pp. 121–130.

[19] A. B. Lewko and B. Waters, "New proof methods for attribute-based encryption: Achieving full security through selective techniques," *IACR Cryptology ePrint Archive*, 2012.

[20] V. Shoup, "Lower bounds for discrete logarithms and related problems," in *Advances in Cryptology*. Berlin, Germany: Springer-Verlag, 1997, pp. 256–266.

[21] D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical identity based encryption with constant size ciphertext," in *Advances in Cryptology*. Berlin, Germany: Springer-Verlag, 2005, pp. 440–456.

[22] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *Advances in Cryptology*. Berlin, Germany: Springer-Verlag, 2008, pp. 146–162.

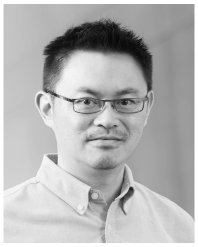
Zhen Liu received the B.Sc. degree in applied mathematics and the M.Sc. degree in mathematics from Shanghai Jiao Tong University, Shanghai, China, and the Ph.D. degrees in computer science from the City University of Hong Kong, Hong Kong, and Shanghai Jiao Tong University.

He is currently a Post-Doctoral Fellow with the Department of Computer Science, City University of Hong Kong. His current interest is applied cryptography, in particular, encryption and signature schemes. With a mathematical background, he is interested in studying provable security and designing cryptographic primitives, for the research problems motivated by practical applications.

Zhenfu Cao (SM'10) received the B.Sc. degree in computer science and technology and the Ph.D. degree in mathematics from the Harbin Institute of Technology, Harbin, China. His research interests mainly include number theory, cryptography, and information security. He has authored over 400 academic papers in leading journals or conferences.

He was early promoted to Associate Professor in 1987, and became a Full Professor in 1991. He is currently a Distinguished Professor with East China Normal University, Shanghai, China. He also serves as a member of the expert panel for the National Nature Science Fund of China.

Prof. Cao is actively involved in the academic community, serving as program cochair or committee member for many international conferences, including the IEEE Global Communications Conference and the IEEE International Conference on Communications (ICC). He is the Associate Editor of *Computers and Security* (Elsevier) and *Security and Communication Networks* (John Wiley), and an Editorial Board Member of *Fundamenta Informaticae* (IOS) and *Peer-to-Peer Networking and Applications* (Springer-Verlag). He also serves as the Guest Editor of several journals, including the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS and *Wireless Communications and Mobile Computing* (Wiley). He was a recipient of a number of awards, including the Youth Research Fund Award of the Chinese Academy of Science (1986), the Ying-Tung Fok Young Teacher Award (1989), the National Outstanding Youth Fund of China (2002), and the Special Allowance by the State Council of China (2004), and a corecipient of the IEEE ICC Best Paper Award (2007). He is also the Principal Investigator of the Asia 3 Foresight Program and the key project of the National Natural Science Foundation of China.



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

He is currently an Associate Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong. His current 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.