Blackbox Traceable CP-ABE: How to Catch People Leaking Their Keys by Selling Decryption Devices on eBay

Zhen Liu[1,2], Zhenfu Cao[1], and Duncan S. Wong[2]

[1] Shanghai Jiao Tong University, Shanghai, China.
zhenliu7@cityu.edu.hk, zfcao@cs.sjtu.edu.cn
[2] City University of Hong Kong, Hong Kong SAR, China.
duncan@cityu.edu.hk

Abstract. In the context of Ciphertext-Policy Attribute-Based Encryption (CP-ABE), if a decryption device associated with an attribute set $S_{\mathcal{D}}$ appears on eBay, and is alleged to be able to decrypt any ciphertexts with policies satisfied by $S_{\mathcal{D}}$, no one including the CP-ABE authorities can identify the malicious user(s) who build such a decryption device using their key(s). This has been known as a major practicality concern in CP-ABE applications, for example, providing fine-grained access control on encrypted data. Due to the nature of CP-ABE, users get decryption keys from authorities associated with attribute sets. If there exists two or more users with attribute sets being the supersets of $S_{\mathcal{D}}$, existing CP-ABE schemes cannot distinguish which user is the malicious one who builds and sells such a decryption device. In this paper, we extend the notion of CP-ABE to support *Blackbox Traceability* and propose a concrete scheme which is able to identify a user whose key has been used in building a decryption device from multiple users whose keys associated with the attribute sets which are all the supersets of $S_{\mathcal{D}}$. The scheme is efficient with sub-linear overhead and when compared with the very recent (non-traceable) CP-ABE scheme due to Lewko and Waters in Crypto 2012, we can consider this new scheme as an extension with the property of *fully collusion-resistant blackbox traceability* added, i.e. an adversary can access an arbitrary number of keys when building a decryption device while the new tracing algorithm can still identify at least one particular key which must have been used for building the underlying decryption device. We show that this new scheme is secure against adaptive adversaries in the standard model, and is highly expressive by supporting any monotonic access structures. Its additional traceability property is also proven against adaptive adversaries in the standard model.
As of independent interest, in this paper, we also consider another scenario which we call it "*found-in-the-wild*". In this scenario, a decryption device is found, for example, from a black market, and reported to an authority (e.g. a law enforcement agency). The decryption device is found to be able to decrypt ciphertexts with certain policy, say $\mathbb{A}$, while the associated attribute set $S_{\mathcal{D}}$ is **missing**. In this found-in-the-wild scenario, we show that the Blackbox Traceable CP-ABE scheme proposed in this paper can still be able to find the malicious users whose keys have been used for building the decryption device, and our scheme can achieve *selective* traceability in the standard model under this scenario.

Keywords: Attribute-Based Encryption; Blackbox Traceability

1 Introduction

Ciphertext-Policy Attribute-Based Encryption (CP-ABE), introduced by Goyal et al. [11], is a versatile one-to-many encryption mechanism 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 possible to find out the identities of all the eligible receivers, and the set of eligible receivers could 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 [23,3] are inefficient to realize the requirement in the example above as they are for one-to-one encryption. Broadcast Encryption (BE) [8] may not be suitable either as the encryptor in BE has to know and specify the exact identities/indices of the receivers. 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 [2,7,10,24,15,20,12,16], progress has been made on the schemes' security, access policy expressivity, and efficiency. In [16], 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 [15,20] have. Specifically, the security proof in [15,20] relies on the *one-use restriction* that a single attribute can only be used once in a policy, and directly extending the schemes in [15,20] to allow attribute reuse would incur significant tradeoffs in efficiency.

One of the major practicality issues of CP-ABE to date is the lacking 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 the 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, while the key has no identity information. As a result, a malicious user, with his attributes shared with multiple other users, might have an intention to leak the corresponding decryption key or some *decryption privilege* in the form of a decryption blackbox/device in which the decryption key is embedded, for example, for financial gain or for some other incentives, as there is little risk of getting caught.

This is an interesting problem in practice as leaking a decryption key or a more advanced decryption device/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 as input, a *tracing algorithm* can find out the user which owns the key. This also includes a scenario that a malicious user sells a new well-formed decryption key for financial gain, and the new decryption key is created from his own key.

Level two is *Blackbox Traceability*, by which given a *decryption blackbox/device*, while the decryption key and even the decryption algorithm could be hidden, the tracing algorithm, which treats the decryption blackbox as an oracle, can still find out the malicious user whose key must have been used in constructing the decryption blackbox.

The problem of building a secure CP-ABE supporting traceability has recently been studied in [18,17,19]. However, as we will review that an *expressive Blackbox* Traceable CP-ABE is yet to be built: (1) the ciphertext access policies in [18,17] only support a single AND gate with wildcard; (2) the traceable CP-ABE in [19] is as fully secure, highly expressive and efficient as a conventional CP-ABE such as the one in [15], but it only supports level one Whitebox Traceability, i.e., it deters malicious users from leaking or selling well-formed decryption keys, but it cannot deter them from selling decryption blackboxes/devices. Below is an example on the importance of achieving Blackbox Traceability.

Key-like Decryption Blackbox for Sale. Using his decryption key (or the decryption keys from multiple colluded malicious users), a malicious user builds a decryption blackbox/device (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 device can decrypt the ciphertext. For simplicity, we call such a decryption blackbox as a ***key-like decryption blackbox***. In practice, such a key-like decryption blackbox could be quite useful and deemed to be very attractive to potential buyers, and the resulting financial gain could be a big incentive for malicious users to build and sell such a blackbox.

1.1 Our Results

In this paper, we propose a new 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 blackbox traceable. Furthermore, this new CP-ABE achieves *fully collusion-resistant* blackbox traceability, that is, the tracing algorithm can find out at least one of the malicious users even if there are an arbitrary number of malicious users colluding by pulling all of their decryption keys together when building a key-like decryption blackbox. Note that collusion-resistant traceability is orthogonal to collusion-resistant security, which is the primary requirement of CP-ABE. *In this paper, traceability is regarded as an additional feature besides the traditional CP-ABE full security, high expressivity and efficiency.*

In addition, the traceability of the scheme is public, that is, anyone can run the tracing algorithm with no additional secret needed. When compared with the most efficient conventional (non-traceable) highly expressive CP-ABE currently available, 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 $\mathcal{K}$, where $\mathcal{K}$ is the number of users in the system, while the private key size and decryption efficiency mainly remain comparable and are independent of the value of $\mathcal{K}$.

To the best of our knowledge, this is the first CP-ABE that simultaneously supports public and fully collusion-resistant blackbox traceability, full security, high expressivity, and without the one-use restriction, and for a system with fully collusion-resistant blackbox traceability, sub-linear overhead is the most efficient one to date. Table 1 compares our scheme with that in [15,16,19] in terms of performance and features (i.e. traceability and one-use restriction), as all the four schemes are fully secure and highly expressive.

In Sec. 2, following the standard definition of conventional CP-ABE, we give a 'functional' definition of CP-ABE, in which we specify a unique index $k \in \{1, \ldots, \mathcal{K}\}$ to each decryption key, that later will enable us to define a tracing algorithm Trace which supports fully collusion-resistant blackbox traceability against key-like decryption blackbox. We call the resulting scheme a Blackbox Traceable CP-ABE (or BT-CP-ABE for short).

On the construction of BT-CP-ABE, instead of building one directly, we first define a simpler primitive called Augmented CP-ABE (or AugCP-ABE for short), then we extend it to BT-CP-ABE. In Sec. 3.1, we define AugCP-ABE as ($\mathsf{Setup_A}$, $\mathsf{KeyGen_A}$, $\mathsf{Encrypt_A}$, $\mathsf{Decrypt_A}$), which is similar to

	Ciphertext Size	Private Key Size	Public Key Size	Pairing Computation in Decryption	Traceability	Without One-Use Restriction
[15]	$2l+2$	$\lvert S\rvert+2$	$\lvert\mathcal{U}\rvert+3$	$2\lvert I\rvert+1$	No	$\times$
[16]	$2l+3$	$\lvert S\rvert+3$	$\lvert\mathcal{U}\rvert+4$	$2\lvert I\rvert+2$	No	$\surd$
[19]	$2l+3$	$\lvert S\rvert+4$	$\lvert\mathcal{U}\rvert+4$	$2\lvert I\rvert+1$	whitebox	$\times$
this work	$2l$ $+17\sqrt{\mathcal{K}}$	$\lvert S\rvert+4$	$\lvert\mathcal{U}\rvert+3$ $+4\sqrt{\mathcal{K}}$	$2\lvert I\rvert+10$	public, blackbox, fully collusion-resistant	$\surd$

[1] All the four schemes are fully secure and highly expressive (i.e. supporting any monotonic access structures).

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

Table 1. Comparison with the conventional CP-ABE in [15,16] and the traceable CP-ABE in [19]

BT-CP-ABE, except that the encryption algorithm $\mathsf{Encrypt_A}(\mathsf{PP}, M, \mathbb{A}, \bar{k})$ takes one more parameter $\bar{k} \in \{1, \ldots, \mathcal{K}+1\}$, and the encrypted message M can be recovered using a decryption key $\mathsf{SK}_{k,S}$, which is identified by $k \in \{1, \ldots, \mathcal{K}\}$ and described by an attribute set S, provided that $(k \geq \bar{k}) \wedge (S \text{ satisfies } \mathbb{A})$, where $\mathbb{A}$ is an access policy. Also, we define the security of AugCP-ABE using message-hiding and (encryption) index-hiding games. In Sec. 3.2, we show how to transform an AugCP-ABE scheme with message-hiding and index-hiding properties to a fully secure BT-CP-ABE scheme. In Sec. 4, we propose an efficient and highly expressive AugCP-ABE scheme, and show that it is message-hiding and index-hiding against adaptively adversaries in the standard model. Combining it with the result in Sec. 3.2, we obtain an efficient, fully secure and highly expressive BT-CP-ABE scheme.

In Sec. 5, we consider the blackbox traceability for another type of decryption blackboxes, which we refer to as ***policy-specific decryption blackbox***. A policy-specific decryption blackbox is a decryption blackbox that is able to decrypt ciphertexts with some specific access policy, say $\mathbb{A}_{\mathcal{D}}$. In other words, unlike a key-like decryption blackbox which has an attribute set, for example, $S_{\mathcal{D}}$ attached as the alleged decryption capability as the blackbox advertised by the seller, a policy-specific decryption blackbox is only known to be able to decrypt ciphertexts with some specific policy $\mathbb{A}_{\mathcal{D}}$. The policy-specific decryption blackbox reflects a different (and possibly more sophisticated) attacking scenario which we call it "*found-in-the-wild.*" In this scenario, a decryption blackbox is found, for example, from a black market, and reported to an authority which could be a law enforcement agency. The decryption blackbox's associated attribute set $S_{\mathcal{D}}$ is missing as it is "found-in-the-wild" while after some testing, it is found that the blackbox can decrypt ciphertexts with certain access policy, say $\mathbb{A}_{\mathcal{D}}$. Interestingly, we show that our BT-CP-ABE scheme is also traceable (where the traceability definition needs to be modified accordingly, the modification is minor) against this policy-specific decryption blackbox, although the traceability can only be proven against selective adversaries.

2 Definitions

We first review the definition of CP-ABE which is based on the work of [15,16] with the exception that in our 'functional' definition, we explicitly assign and identify users using unique indices, and let $\mathcal{K}$ be the number of users in a CP-ABE system. Then we introduce the traceability definition against key-like decryption blackbox. Predefining the number of users is indeed a weakness as well as a necessary cost for achieving blackbox traceability, but we stress that in practice, this should not incur much concern, and all the existing blackbox traceable systems (e.g. [14,5,6,9]) have the same setting. Also being consistent with the conventional definition of CP-ABE [15,16], the user indices are not used in normal encryption (i.e. the encryptors do not need to know the indices of any users in order to encrypt) and different users (with different indices) may have the same attribute set.

2.1 CP-ABE and Security Models

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

$\mathsf{Setup}(\lambda, \mathcal{U}, \mathcal{K}) \to (\mathsf{PP}, \mathsf{MSK})$. 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.

$\mathsf{KeyGen}(\mathsf{PP}, \mathsf{MSK}, S) \to \mathsf{SK}_{k,S}$. 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 $k \in \{1, \ldots, \mathcal{K}\}$.

$\mathsf{Encrypt}(\mathsf{PP}, M, \mathbb{A}) \to CT$. 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.

$\mathsf{Decrypt}(\mathsf{PP}, CT, \mathsf{SK}_{k,S}) \to M$ 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.

Now we define the security of a CP-ABE system using a message-hiding game, which is a typical semantic security game and is based on that for conventional CP-ABE [15,16] security against adaptive adversaries, except that each key is identified by a unique index. 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., to query a private decryption key for an attribute set S, the adversary submits (k, S) to the challenger, where k is the index to be assigned to the corresponding decryption key.

It is worth noticing that: (1) for clarity, 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.

$\mathsf{Game}_{\mathsf{MH}}$. 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 clear that a secure CP-ABE system defined as above has all the appealing properties that a conventional CP-ABE system [15,16] has, that is, fully collusion-resistant security, fine-grained access control on encrypted data, and efficient one-to-many encryption.

2.2 BT-CP-ABE: Traceability

Now we define the traceability against key-like decryption blackbox, and call the new system a Blackbox Traceable CP-ABE (or BT-CP-ABE for short). Our definition is loosely related to the traitor tracing feature in broadcast encryption [6,9]. A key-like decryption blackbox $\mathcal{D}$ in our setting is viewed as a probabilistic circuit that takes as input a ciphertext CT and outputs a message M or $\perp$, and such a decryption blackbox does not need to be perfect, namely, we only require it to be able to decrypt with non-negligible success probability. *In particular, the 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 valid 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 who are interested in decrypting ciphertexts with access policies which can be satisfied by $S_{\mathcal{D}}$. We now define a tracing algorithm as follows.

$\mathsf{Trace}^{\mathcal{D}}(\mathsf{PP}, S_{\mathcal{D}}, \epsilon) \rightarrow \mathbb{K}_T \subseteq \{1, \ldots, \mathcal{K}\}$. *This 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 Tracing Game captures the notion of **fully collusion-resistant traceability**. In the game, the adversary targets to build a decryption blackbox $\mathcal{D}$ that functions as a private decryption key with attribute set $S_{\mathcal{D}}$ (as the name of key-like decryption blackbox implies). The tracing algorithm, on the other side, is designed to extract the index of at least one of the malicious users whose decryption keys have been used for constructing $\mathcal{D}$.

$\mathsf{Game}_{\mathsf{TR}}$. The 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 the challenger responds with $\mathsf{SK}_{k_i, S_{k_i}}$.
(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 $\mathsf{Trace}^{\mathcal{D}}(\mathsf{PP}, S_{\mathcal{D}}, \epsilon)$ 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 $\mathsf{TRAdv}_{\mathcal{A}}$ the probability that adversary $\mathcal{A}$ wins this game.

Definition 2. *A $\mathcal{K}$-user Blackbox Traceable CP-ABE system is traceable if for all polynomial-time adversaries $\mathcal{A}$ the advantage $\mathsf{TRAdv}_{\mathcal{A}}$ is negligible in λ.*

Remark: For a useful key-like decryption blackbox $\mathcal{D}$, the traced $\mathbb{K}_T$ must satisfy $(\mathbb{K}_T \neq \emptyset) \wedge (\mathbb{K}_T \subseteq \mathbb{K}_{\mathcal{D}}) \wedge (\exists k_t \in \mathbb{K}_T\ s.t.\ S_{k_t} \supseteq S_{\mathcal{D}})$ for traceabililty. (1) $(\mathbb{K}_T \neq \emptyset) \wedge (\mathbb{K}_T \subseteq \mathbb{K}_{\mathcal{D}})$ captures the preliminary traceability that the tracing algorithm can extract at least one malicious user and the coalition of malicious users cannot frame any innocent user. Note that such a preliminary traceability is a *weak traceability* that may not be useful enough in practice. Specifically, consider a key-like decryption blackbox $\mathcal{D}$ built from the private decryption keys of users k_1 and k_2 who were authorized high-value attribute set S_{k_1} and low-value attribute set S_{k_2}, respectively, and assume that $S_{k_2} \not\supseteq S_{k_1}$ and the decryption ability of $\mathcal{D}$ is described by $S_{\mathcal{D}} = S_{k_1}$, e.g., $S_{\mathcal{D}} = S_{k_1} = \{\text{Senior Manager}\}$, and $S_{k_2} = \{\text{Intern}\}$. A scheme is considered to be weak traceable if its Trace algorithm only extracts k_2 from $\mathcal{D}$ as the malicious user. This may not be satisfactory in practice as $\mathcal{D}$ having the decryption ability of attribute set $\{\text{Senior Manager}\}$ implies that there must be some user having attribute "Senior Manager" participated in building $\mathcal{D}$ yet the algorithm was only able to trace $\mathcal{D}$ to an "Intern", who has less to lose. (2) $(\exists k_t \in \mathbb{K}_T\ s.t.\ S_{k_t} \supseteq S_{\mathcal{D}})$ captures *strong traceability* that the Trace 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 $S_{\mathcal{D}}$. As a related work, comparable weak and strong traceability notions in the setting of predicate encryption were considered in [14]. In this paper we focus on the strong traceability of CP-ABE, and unless stated otherwise, by the *traceability* we mean the *strong traceability.*

Note that the tracing game above does not limit the number of colluded users. Also note that, as of [5,6,9,14], we are modeling a stateless (resettable) decryption blackbox – the decryption blackbox is just an oracle and maintains no state between activations.

3 Augmented CP-ABE

Following the routes of [5,6,9], instead of constructing a BT-CP-ABE directly, we define a simpler primitive called Augmented CP-ABE (or AugCP-ABE for short) and its security notions first, then we show that a secure AugCP-ABE can be transformed to a secure and traceable BT-CP-ABE scheme. In Sec. 4, we propose a concrete construction of AugCP-ABE.

3.1 Definitions

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

$\mathsf{Setup_A}(\lambda, \mathcal{U}, \mathcal{K}) \to (\mathsf{PP}, \mathsf{MSK})$. 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.

$\mathsf{KeyGen_A}(\mathsf{PP}, \mathsf{MSK}, S) \to \mathsf{SK}_{k,S}$. The algorithm takes as input PP, the master secret key MSK, and an attribute set S, and outputs a private key $\mathsf{SK}_{k,S}$, which is assigned and identified by a unique index $k \in \{1, \ldots, \mathcal{K}\}$.

$\mathsf{Encrypt_A}(\mathsf{PP}, M, \mathbb{A}, \bar{k}) \to CT$. 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**.

$\mathsf{Decrypt_A}(\mathsf{PP}, CT, \mathsf{SK}_{k,S}) \to M$ or $\perp$. The algorithm takes as input 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.

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 $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup_A}(\lambda, \mathcal{U}, \mathcal{K})$, $\mathsf{SK}_{k,S} \leftarrow \mathsf{KeyGen_A}(\mathsf{PP}, \mathsf{MSK}, S)$, $CT \leftarrow \mathsf{Encrypt_A}(\mathsf{PP}, M, \mathbb{A}, \bar{k})$. If ($S$ satisfies $\mathbb{A}$) $\wedge (k \geq \bar{k})$ then $\mathsf{Decrypt_A}(\mathsf{PP}, CT, \mathsf{SK}_{k,S}) = M$.

It is worth noticing that during decryption if the attribute set S of a private decryption key satisfies the access policy $\mathbb{A}$ of a ciphertext, the decryption works, regardless of the value of key index k or encryption index $\bar{k}$, but whether the output message is equal to the encrypted message is determined by the values 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 always set $\bar{k} = 1$, then the functions of AugCP-ABE are identical to that of BT-CP-ABE. Actually, the idea behind converting an AugCP-ABE scheme to a BT-CP-ABE scheme is to construct an AugCP-ABE scheme with (encryption) index-hiding property, and then always set $\bar{k} = 1$ in normal encryption, while use $\bar{k} \in \{1, \ldots, \mathcal{K}+1\}$ in generating ciphertexts for tracing.

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 $\mathsf{Setup_A}(\lambda, \mathcal{U}, \mathcal{K})$ 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 $CT \leftarrow \mathsf{Encrypt_A}(\mathsf{PP}, M_b, \mathbb{A}^*, \bar{k})$ to $\mathcal{A}$.

Phase 2. For $i = q_1 + 1$ to q, $\mathcal{A}$ adaptively submits (k_i, S_{k_i}), and the challenger responds with $\mathsf{SK}_{k_i, S_{k_i}}$.

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 $CT \leftarrow \mathsf{Encrypt_A}(\mathsf{PP}, M_b, \mathbb{A}^*, 1)$ 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 $CT \leftarrow \mathsf{Encrypt_A}(\mathsf{PP}, M_b, \mathbb{A}^*, \mathcal{K}+1)$ 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 3. *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**, for any non-empty attribute set $S^* \subseteq \mathcal{U}$, we define **the strictest access policy** as $\mathbb{A}_{S^*} = \bigwedge_{x \in S^*} x$, and require that an adversary cannot distinguish between an encryption using $(\mathbb{A}_{S^*}, \bar{k})$ and $(\mathbb{A}_{S^*}, \bar{k}+1)$ without a private decryption key $\mathsf{SK}_{\bar{k}, S_{\bar{k}}}$ where $S_{\bar{k}} \supseteq S^*$. 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 the challenger responds with $\mathsf{SK}_{k_i, S_{k_i}}$.
Challenge. $\mathcal{A}$ submits a message M and a non-empty attribute set S^*. The challenger flips a random coin $b \in \{0, 1\}$, and sends $CT \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, \mathbb{A}_{S^*}, \bar{k} + b)$ 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 $\{(k_i, S_{k_i})\}_{1 \le i \le q}$ can satisfy $(k_i = \bar{k}) \wedge (S_{k_i}$ satisfies $\mathbb{A}_{S^*})$, i.e. $(k_i = \bar{k}) \wedge (S_{k_i} \supseteq S^*)$. 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 4. *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.*

3.2 Reducing BT-CP-ABE to AugCP-ABE

We now show that an AugCP-ABE with message-hiding and index-hiding implies a secure and traceable BT-CP-ABE. Let $\Sigma_{\mathsf{A}} = (\mathsf{Setup}_{\mathsf{A}}, \mathsf{KeyGen}_{\mathsf{A}}, \mathsf{Encrypt}_{\mathsf{A}}, \mathsf{Decrypt}_{\mathsf{A}})$ be an AugCP-ABE with message-hiding and index-hiding, define $\mathsf{Encrypt}(\mathsf{PP}, M, \mathbb{A}) = \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, \mathbb{A}, 1)$, then $\Sigma = (\mathsf{Setup}_{\mathsf{A}}, \mathsf{KeyGen}_{\mathsf{A}}, \mathsf{Encrypt}, \mathsf{Decrypt}_{\mathsf{A}})$ is a BT-CP-ABE derived from Σ_{A}, and the tracing algorithm is defined as

$\mathsf{Trace}^{\mathcal{D}}(\mathsf{PP}, S_{\mathcal{D}}, \epsilon) \to \mathbb{K}_T \subseteq \{1, \ldots, \mathcal{K}\}$: Given a key-like decryption blackbox $\mathcal{D}$ associated with a non-empty attribute set $S_{\mathcal{D}}$ and probability $\epsilon > 0$, the tracing algorithm works as follows: [3]

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 $CT \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, \mathbb{A}_{S_{\mathcal{D}}}, k)$, where $\mathbb{A}_{S_{\mathcal{D}}}$ is the strictest access policy of $S_{\mathcal{D}}$.
      iii. Query oracle $\mathcal{D}$ on input CT which contains $\mathbb{A}_{S_{\mathcal{D}}}$, 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 $\hat{p}_k - \hat{p}_{k+1} \ge \epsilon/(4\mathcal{K})$. Then output $\mathbb{K}_T$ as the index set of the private decryption keys of malicious users.

Remark: Note that the *strictest access policy* used in the index-hiding game $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$ and the tracing algorithm Trace does not impose any limitation to traceable CP-ABE. Instead, it is an efficient way to ensure that the traced malicious users are the reasonable suspects. As a key-like decryption blackbox $\mathcal{D}$ is advertised that it functions like a private decryption key with attribute set $S_{\mathcal{D}}$, a ciphertext associated with the strictest access policy $\mathbb{A}_{S_{\mathcal{D}}}$ will be decrypted by $\mathcal{D}$ accordingly. Although it might look more appealing to have the index-hiding property for any access policy, the following Theorem 1 shows that the strictest access policy is sufficient for ensuring the traceability against key-like decryption blackbox for the derived BT-CP-ABE scheme.

[3] The tracing algorithm uses a technique based on that in broadcast encryption by [5,6,9].

Theorem 1. *If Σ_{A} is an AugCP-ABE with message-hiding and index-hiding properties, then Σ is a secure and traceable BT-CP-ABE.*

Proof. Note that Σ is a special case of Σ_{A} where the encryption algorithm always sets $\bar{k} = 1$. Hence, $\mathsf{Game}_{\mathsf{MH}}$ for Σ is identical to $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$ for Σ_{A}, which implies that $\mathsf{MHAdv}_{\mathcal{A}}$ for Σ in $\mathsf{Game}_{\mathsf{MH}}$ is equal to $\mathsf{MH}^{\mathsf{A}}_1\mathsf{Adv}_{\mathcal{A}}$ for Σ_{A} in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$, i.e., if Σ_{A} is message-hiding (in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$), then Σ is secure.

Now we show that if Σ_{A} is message-hiding (in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_{\mathcal{K}+1}}$) and index-hiding, Σ is traceable. In the proof sketch below, which is based on that of [5,6,9], we show that if the key-like decryption blackbox output by the adversary is a useful one then the traced $\mathbb{K}_T$ will satisfy $(\mathbb{K}_T \neq \emptyset) \wedge (\mathbb{K}_T \subseteq \mathbb{K}_{\mathcal{D}}) \wedge (\exists k_t \in \mathbb{K}_T \ s.t. \ S_{k_t} \supseteq S_{\mathcal{D}})$ with overwhelming probability, which implies that the adversary can win the game $\mathsf{Game}_{\mathsf{TR}}$ only with negligible probability, i.e., $\mathsf{TRAdv}_{\mathcal{A}}$ is negligible.

Let $\mathcal{D}$ be the key-like decryption blackbox output by the adversary, and $S_{\mathcal{D}}$ be the attribute set describing $\mathcal{D}$. Define

$$p_{\bar{k}} = \Pr[\mathcal{D}(\mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, \mathbb{A}_{S_{\mathcal{D}}}, \bar{k})) = M],$$

where the probability is taken over the random choice of message M and the random coins of $\mathcal{D}$. We have that $p_1 \geq \epsilon$ and $p_{\mathcal{K}+1}$ is negligible. The former follows the fact that $\mathcal{D}$ is a useful key-like decryption blackbox, and the later follows that Σ_{A} is message-hiding (in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_{\mathcal{K}+1}}$). Then there must exist some $k \in \{1, \ldots, \mathcal{K}\}$ such that $p_k - p_{k+1} \geq \epsilon/(2\mathcal{K})$. By the Chernoff bound it follows that with overwhelming probability, $\hat{p}_k - \hat{p}_{k+1} \geq \epsilon/(4\mathcal{K})$. Hence, we have $\mathbb{K}_T \neq \emptyset$.

For any $k \in \mathbb{K}_T$ (i.e., $\hat{p}_k - \hat{p}_{k+1} \geq \frac{\epsilon}{4\mathcal{K}}$), we know, by Chernoff, that with overwhelming probability $p_k - p_{k+1} \geq \epsilon/(8\mathcal{K})$. Clearly $(k \in \mathbb{K}_{\mathcal{D}}) \wedge (S_k \supseteq S_{\mathcal{D}})$ since otherwise, $\mathcal{D}$ can be directly used to win the index-hiding game for Σ_A. Hence, we have $(\mathbb{K}_T \subseteq \mathbb{K}_{\mathcal{D}}) \wedge (S_{\mathcal{D}} \subseteq S_k \ \forall k \in \mathbb{K}_T)$.

4 An Efficient Augmented CP-ABE

Now we construct an AugCP-ABE scheme that is as secure and expressive as the CP-ABE scheme in [16]. To obtain traceability in the derived BT-CP-ABE scheme we will use the standard tracing techniques which were used by [5,6,9] in the setting of broadcast encryption. The challenge is to apply the tracing techniques to the setting of CP-ABE *securely and efficiently*.

4.1 Preliminaries

Before proposing a concrete construction for AugCP-ABE, we first review some preliminaries.

Linear Secret-Sharing Schemes. As of previous work, we use linear secret-sharing schemes (LSSS) to realize monotonic access structures which specify the access policies associated with ciphertexts. The formal definitions of access structures and LSSS can be found in [24,15,16]. Informally, an LSSS is a share-generating matrix A whose rows $\{A_i\}$ are labeled by attributes through a function ρ. When we consider the column vector $\boldsymbol{v} = (s, r_2, \ldots, r_n)$, 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, $A\boldsymbol{v}$ is the vector of l shares of the secret s, and the share $\lambda_i = (A\boldsymbol{v})_i$, i.e. the inner product $A_i \cdot \boldsymbol{v}$, belongs to attribute $\rho(i)$. A user's 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.

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 $N = p_1p_2p_3$, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ 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 $\mathbb{G}_T$). More details can be found in [15,16].

Complexity Assumptions. The message-hiding property of our AugCP-ABE scheme will rely on four assumptions (the Assumption 1 in [16], 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), which are used in [16] to achieve full security of their CP-ABE scheme while eliminating the one-use restriction. The index-hiding property will rely on two assumptions (3-Party Diffie-Hellman Assumption and Decisional Linear Assumption) that are used in [9] to achieve traceability in the setting of broadcast encryption. We refer to [16,9] for the details of these assumptions.

Notations. Suppose the number of users $\mathcal{K}$ in the system equals m^2 for some m [4]. 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 positive integer, say m, by $[m]$ we mean the set $\{1, 2, \ldots, m\}$. For a given vector $\boldsymbol{v} = (v_1, \ldots, v_d)$, by $g^{\boldsymbol{v}}$ we mean the vector $(g^{v_1}, \ldots, g^{v_d})$. Furthermore, for $g^{\boldsymbol{v}} = (g^{v_1}, \ldots, g^{v_d})$ and $g^{\boldsymbol{w}} = (g^{w_1}, \ldots, g^{w_d})$, by $g^{\boldsymbol{v}} \cdot g^{\boldsymbol{w}}$ we mean the vector $(g^{v_1+w_1}, \ldots, g^{v_d+w_d})$, i.e. $g^{\boldsymbol{v}} \cdot g^{\boldsymbol{w}} = g^{\boldsymbol{v}+\boldsymbol{w}}$, and by $e_d(g^{\boldsymbol{v}}, g^{\boldsymbol{w}})$ we mean $\prod_{k=1}^{d} e(g^{v_k}, g^{w_k})$, i.e. $e_d(g^{\boldsymbol{v}}, g^{\boldsymbol{w}}) = \prod_{k=1}^{d} e(g^{v_k}, g^{w_k}) = e(g, g)^{(\boldsymbol{v} \cdot \boldsymbol{w})}$ where $(\boldsymbol{v} \cdot \boldsymbol{w})$ is the inner product of $\boldsymbol{v}$ and $\boldsymbol{w}$. Given a bilinear group order N, one can randomly choose $r_x, r_y, r_z \in \mathbb{Z}_N$, and set $\boldsymbol{\chi}_1 = (r_x, 0, r_z)$, $\boldsymbol{\chi}_2 = (0, r_y, r_z)$, $\boldsymbol{\chi}_3 = \boldsymbol{\chi}_1 \times \boldsymbol{\chi}_2 = (-r_yr_z, -r_xr_z, r_xr_y)$. Let $\mathit{span}\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$ be the subspace spanned by $\boldsymbol{\chi}_1$ and $\boldsymbol{\chi}_2$, i.e. $\mathit{span}\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\} = \{\nu_1\boldsymbol{\chi}_1 + \nu_2\boldsymbol{\chi}_2 | \nu_1, \nu_2 \in \mathbb{Z}_N\}$. We can see that $\boldsymbol{\chi}_3$ is orthogonal to the subspace $\mathit{span}\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$ and $\mathbb{Z}_N^3 = \mathit{span}\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2, \boldsymbol{\chi}_3\} = \{\nu_1\boldsymbol{\chi}_1 + \nu_2\boldsymbol{\chi}_2 + \nu_3\boldsymbol{\chi}_3 | \nu_1, \nu_2, \nu_3 \in \mathbb{Z}_N\}$. For any $\boldsymbol{v} \in \mathit{span}\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$, we have $(\boldsymbol{\chi}_3 \cdot \boldsymbol{v}) = 0$, and for random $\boldsymbol{v} \in \mathbb{Z}_N^3$, $(\boldsymbol{\chi}_3 \cdot \boldsymbol{v}) \neq 0$ occurs with overwhelming probability.

4.2 Our Approach

Note that the Traitor Tracing schemes in broadcast encryption [5,6,9] achieved fully collusion-resistant blackbox traceability at the cost of sub-linear overhead, which is the most efficient level to date. It will be tempting to try in a straightforward way to combine such a Traitor Tracing system and a CP-ABE for obtaining a BT-CP-ABE. However, the resulting system cannot achieve the desired security (i.e. *strong* traceability). Consider the following (misguided) approach. Suppose that we created both a CP-ABE and a Traitor Tracing system each for $\mathcal{K}$ users, where each user

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

has the same index in both systems. To encrypt a message M, an algorithm splits the message randomly into two pieces M_P and M_I such that $M_P \cdot M_I = M$, then encrypts M_P under CP-ABE and M_I under the Traitor Tracing system. To decrypt, we need to decrypt under both systems. However, such an approach can only provide weak traceability[5]. In particular, if two users, Alice with attribute set S_A in CP-ABE and index k_A in both systems, and Bob with attribute set S_B in CP-ABE and index k_B in both systems, collude to make a decryption blackbox $\mathcal{D}$ with attribute set $S_{\mathcal{D}} \subseteq S_A$, while $S_B \cap S_A = \emptyset$. The blackbox uses Alice's key (the part corresponding to S_A) to decrypt the ciphertext from the CP-ABE system and Bob's key (the part corresponding to k_B) to decrypt the ciphertext from the Traitor Tracing system. The tracing algorithm would identify Bob as a malicious user, but S_B is uncorrelated to $S_{\mathcal{D}}$.

The idea behind the techniques of achieving **strong** traceability is to set a user's private decryption key such that it must be *simultaneously* used in both CP-ABE and the Tracing part in a BT-CP-ABE. Boneh and Waters [6] handled a similar situation where they intertwined a broadcast encryption scheme [4] and a Traitor Tracing scheme [5] to build an Augmented Broadcast Encryption (AugBE) scheme. Inspired by their approach, we tried to intertwine a CP-ABE [16] and a Traitor Tracing system [9] to build an AugCP-ABE scheme. The obstacle comes from the setting that in CP-ABE the decryption privilege of a user is determined by his attributes rather than by his index as in broadcast encryption. In particular, in AugBE [6,9], the indices of users are simultaneously used to determine users' decryption privilege (for broadcast encryption part) and to identify users (for Tracing), and the construction and the proof of index-hiding of AugBE [6,9] are based on this fact. In contrast, in BT-CP-ABE, the attributes are used to determine users' decryption privilege (for CP-ABE part) while the indices are used to identify users (for Tracing part), and to intertwine the two essentially uncorrelated parts, we need new ideas and techniques.

A straightforward combination will result in schemes that are either not provable or inefficient with ciphertext of size $O(\sqrt{\mathcal{K}} \cdot |\mathbb{A}|)$ where $|\mathbb{A}|$ is the size of an access policy. In the following, based on the CP-ABE in [16] with our particular designs and contructions, we propose a *secure* AugCP-ABE which is also *efficient* with ciphertext of size $O(\sqrt{\mathcal{K}} + |\mathbb{A}|)$.

4.3 AugCP-ABE Construction

$\mathsf{Setup_A}(\lambda, \mathcal{U}, \mathcal{K} = m^2) \to (\mathsf{PP}, \mathsf{MSK})$. Let $\mathbb{G}$ be a bilinear group of order $N = p_1p_2p_3$ (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.

[5] A similar approach was used in [14] to introduce weak traceability to predicate encryption.

$\mathsf{KeyGen}_{\mathsf{A}}(\mathsf{PP}, \mathsf{MSK}, S) \to \mathsf{SK}_{(i,j),S}$. The algorithm first sets $ctr = ctr + 1$ and computes the corresponding index in the form of (i, j) where $1 \le i, j \le m$ 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 $\mathsf{SK}_{(i,j),S} =$

$$\Big(\begin{aligned} & K_{i,j} = g^{\alpha_i} g^{r_i c_j} f^{\sigma_{i,j}} h^{\delta_{i,j}} R, \\ & K'_{i,j} = g^{\sigma_{i,j}} R', \ K''_{i,j} = g^{\delta_{i,j}} R'', \ K'''_{i,j} = Z_i^{\sigma_{i,j}} R''', \\ & \{K_{i,j,x} = U_x^{\sigma_{i,j}} R_x\}_{x \in S} \end{aligned} \Big).$$

The value of (i, j) is implicitly contained in $\mathsf{SK}_{(i,j),S}$.

$\mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, \mathbb{A} = (A, \rho), (\bar{i}, \bar{j})) \to CT$. 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

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

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

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

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

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

$$\begin{aligned} & \boldsymbol{R}_i = g^{\boldsymbol{v}_i}, \ \boldsymbol{R}'_i = g^{\kappa \boldsymbol{v}_i}, \\ & Q_i = g^{s_i}, \ Q'_i = f^{s_i} Z_i^{t_i} f^{\pi}, \ Q''_i = h^{s_i}, \ Q'''_i = g^{t_i}, \\ & T_i = E_i^{\hat{s}_i}. \end{aligned}$$

   - if $i \ge \bar{i}$: sets

$$\begin{aligned} & \boldsymbol{R}_i = G_i^{s_i \boldsymbol{v}_i}, \ \ \boldsymbol{R}'_i = G_i^{\kappa s_i \boldsymbol{v}_i}, \\ & Q_i = g^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}, \ \ Q'_i = f^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)} Z_i^{t_i} f^{\pi}, \ \ Q''_i = h^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}, \ \ Q'''_i = g^{t_i}, \\ & T_i = M \cdot E_i^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}. \end{aligned}$$

2. For each column $j \in [m]$:
   - if $j < \bar{j}$: randomly chooses $\mu_j \in \mathbb{Z}_N$, and sets $\boldsymbol{C}_j = H_j^{\tau(\boldsymbol{v}_c + \mu_j \boldsymbol{\chi}_3)} \cdot g^{\kappa \boldsymbol{w}_j}, \ \ \boldsymbol{C}'_j = g^{\boldsymbol{w}_j}$.
   - if $j \ge \bar{j}$: sets $\boldsymbol{C}_j = H_j^{\tau \boldsymbol{v}_c} \cdot g^{\kappa \boldsymbol{w}_j}, \ \ \boldsymbol{C}'_j = g^{\boldsymbol{w}_j}$.
3. For each $k \in [l]$: sets $P_k = f^{A_k \cdot \boldsymbol{u}} U_{\rho(k)}^{-\xi_k}, \ \ P'_k = g^{\xi_k}$.

$\mathsf{Decrypt_A}(\mathsf{PP}, CT, \mathsf{SK}_{(i,j),S}) \to M$ or $\perp$. The algorithm parses CT to $CT = \langle (A,\rho), (\boldsymbol{R}_i, \boldsymbol{R}'_i, Q_i, Q'_i, Q''_i, Q'''_i, T_i)_{i=1}^m, (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^m, (P_k, P'_k)_{k=1}^l \rangle$. If S does not satisfy (A,ρ), the algorithm outputs $\perp$, 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

$$D_P = \prod_{\rho(k)\in S} \left(e(K'_{i,j}, P_k) e(K_{i,j,\rho(k)}, P'_k)\right)^{\omega_k}$$
$$= \prod_{\rho(k)\in S} \left(e(g^{\sigma_{i,j}}, f^{A_k \cdot \boldsymbol{u}})\right)^{\omega_k} = e(g^{\sigma_{i,j}}, f)^{\pi}.$$

2. Computes $D_I = \frac{e(K_{i,j},Q_i)\cdot e(K'''_{i,j},Q'''_i)}{e(K'_{i,j},Q'_i)\cdot e(K''_{i,j},Q''_i)} \cdot \frac{e_3(\boldsymbol{R}'_i,\boldsymbol{C}'_j)}{e_3(\boldsymbol{R}_i,\boldsymbol{C}_j)}$.
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. The correctness details can be found in Appendix A.

Remark: In $\mathsf{Encrypt_A}$, π is the secret shared according to the LSSS (A,ρ), and is for generating ciphertext components $(P_k, P'_k)_{k=1}^l$, so that only users with eligible attribute sets can recover $D_P = e(g,f)^{\pi\sigma_{i,j}}$. To intertwine the CP-ABE part and Tracing part, f^{π} is embedded in Q'_i, i.e., in Tracing ciphertext components, although $(\boldsymbol{R}_i, \boldsymbol{R}'_i, Q_i, T_i, \boldsymbol{C}_j, \boldsymbol{C}'_j)$ are the same as that of [9], Q'_i is different and Q''_i and Q'''_i are new components we introduced. We stress that $Z_i^{t_i}$ (in Q'_i) is the crucial component that intertwines the Tracing part (i.e. $f^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)}$ for $i \geq \bar{i}$ and f^{s_i} for $i < \bar{i}$) and the CP-ABE part (i.e. f^{π}) *securely and efficiently.* In a straightforward combination without $Z_i^{t_i}$ (i.e. $Q'_i = f^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)} f^{\pi}$ for $i \geq \bar{i}$ and $Q'_i = f^{s_i} f^{\pi}$ for $i < \bar{i}$), the index-hiding property will be hard to prove, and to obtain provable index-hiding, different π_i has to be used for different i (i.e. $Q'_i = f^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)} f^{\pi_i}$ for $i \geq \bar{i}$ and $Q'_i = f^{s_i} f^{\pi_i}$ for $i < \bar{i}$), but this will make the CP-ABE part have ciphertext size of $O(\sqrt{\mathcal{K}} \cdot l)$, rather than $O(l)$ as above. The using of $Z_i^{t_i}$ (and the introduction of Q'_i, Q''_i and Q'''_i) enables us to prove the index-hiding property while achieving (efficient) ciphertext size of $O(\sqrt{\mathcal{K}} + l)$. In particular, when reducing the index-hiding property to the 3-Party Diffie-Hellman assumption, $f^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)}$ and f^{π} will contain terms of g^{bc} and g^{ac} respectively, where the simulator cannot compute, and only with the help of $Z_i^{t_i}$ the simulator can cancel them and form the challenge ciphertext, i.e., let Z_i contain the term g^c and t_i contain the terms b and a, so that the terms g^{bc} and g^{ac} in $f^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)}$ and f^{π} can be canceled out by those in $Z_i^{t_i}$, while $Q'''_i = g^{t_i}$ can be formed using terms $A = g^a$ and $B = g^b$. Details are given in Appendix B.2, i.e. the proof of Lemma 1.

4.4 AugCP-ABE Security

The following Theorem 2 and 3 prove that our AugCP-ABE scheme is message-hiding, and Theorem 4 prove that our AugCP-ABE scheme is index-hiding.

Theorem 2. *Under the 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.*

Proof. Note that the structures of CP-ABE part of our AugCP-ABE scheme are similar to that of the CP-ABE scheme in [16], the proof of Theorem 2 is also similar to that of [16]. For simplicity, here we prove the theorem by reducing the message-hiding property of our AugCP-ABE scheme in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$ to the security of CP-ABE scheme in [16]. The proof details can be found in Appendix B.1.

Theorem 3. *No polynomial time adversary can win the game* $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_{\mathcal{K}+1}}$ *with non-negligible advantage.*

Proof. The argument for message-hiding in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_{\mathcal{K}+1}}$ is very straightforward since an encryption to index $\mathcal{K}+1 = (m+1, 1)$ contains no information about the message. The simulator simply runs actual $\mathsf{Setup}_{\mathsf{A}}$ and $\mathsf{KeyGen}_{\mathsf{A}}$ algorithms and encrypts the message M_b by the challenge access policy $\mathbb{A}^*$ and index $(m+1, 1)$. Since for all $i = 1$ to m, the values of $T_i = E_i^{\hat{s}_i}$ contains no information about the message, the bit b is perfectly hidden and $\mathsf{MH}^{\mathsf{A}}_{\mathcal{K}+1}\mathsf{Adv}_{\mathcal{A}} = 0$.

Theorem 4. *Suppose that the 3-Party Diffie-Hellman Assumption in a Subgroup (defined in [16]), the 3-Party Diffie-Hellman Assumption (defined in [9]) and the Decisional Linear Assumption hold. Then no polynomial time adversary can win* $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$ *with non-negligible advantage.*

Proof. Theorem 4 follows from the following Lemma 1 and Lemma 2 immediately.

Lemma 1. *Suppose that the 3-Party Diffie-Hellman Assumption in a Subgroup holds. Then no polynomial time adversary can 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.*

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 a non-empty attribute set S^*. There is not any restriction on S^*.

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 a non-empty attribute set S^* with the restriction that $S_{(\bar{i},\bar{j})}$ does not satisfy the corresponding strictest access policy $\mathbb{A}_{S^*}$ (i.e. $S^* \setminus S_{(\bar{i},\bar{j})} \neq \emptyset$).

The **Case I** is easy to handle using the similar proof ideas in [9] as the adversary will not query a private key with the challenge index $(\bar{i}, \bar{j})$. The **Case II** captures the index-hiding requirement for CP-ABE in that even if a user has a key with index $(\bar{i}, \bar{j})$ he cannot distinguish between an encryption to $(\mathbb{A}_{S^*}, (\bar{i}, \bar{j}))$ and $(\mathbb{A}_{S^*}, (\bar{i}, \bar{j}+1))$ if the corresponding attribute set $S_{(\bar{i},\bar{j})}$ does not satisfies $\mathbb{A}_{S^*}$. This is the most challenging part of achieving the strong traceability in CP-ABE securely and efficiently, and our particular construction (of the crucial components $Z_i^{t_i}$ (in Q'_i) and $Q'''_i = g^{t_i}$ in the ciphertext) is driven by and serves this aim. These ciphertext components are crucial for the proof to use the underlying assumption to simulate the real attack game when $\mathcal{A}$ behaves in **Case II**. The proof details of Lemma 1 can be found in Appendix B.2.

Lemma 2. *Suppose that the 3-Party Diffie-Hellman Assumption in a Subgroup (defined in [16]), the 3-Party Diffie-Hellman Assumption (defined in [9]) and the Decisional Linear Assumption hold. Then no polynomial time adversary can 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 [9], to prove this lemma we define the following hybrid experiments: 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. *Suppose that the 3-Party Diffie-Hellman Assumption in a Subgroup holds. Then no polynomial time adversary can distinguish between experiments H_1 and H_2 with non-negligible advantage.*

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. The indistinguishability of H_2 and H_3 can be proved using a proof similar to that of Lemma 6.3 in [9], which was used to prove the indistinguishability of similar hybrid experiments for their Augmented Broadcast Encryption (AugBE) scheme. For simplicity, we will prove Claim 2 by a reduction from our AugCP-ABE scheme to the AugBE scheme in [9]. The proof details can be found in Appendix B.3.

5 Policy-Specific Decryption Blackbox

In previous sections, we considered the traceability against key-like decryption blackboxes, which allow a seller (on eBay) to advertise the alleged decryption privilege of a blackbox $\mathcal{D}$ by an attribute set $S_{\mathcal{D}}$. The seller can claim that $\mathcal{D}$ can decrypt a ciphertext (with at least a non-negligible probability) if the ciphertext access policy $\mathbb{A}$ is satisfied by $S_{\mathcal{D}}$. Note that $\mathbb{A}$ can be any arbitrary access policy as long as it can be satisfied by $S_{\mathcal{D}}$. This type of decryption blackboxes are very powerful and therefore, could also be one of the most crucial issues to solve in practice using a blackbox tracing algorithm. In this section, we focus our attention on another interesting scenario which requires us to deal with another type of decryption blackboxes, which we call it a *policy-specific decryption blackbox*.

Policy-Specific Decryption Blackbox on Sale. Attempting to invalidate the possible tracing algorithm (such as the one we proposed above), a malicious user may build and sell a decryption blackbox which decrypts ciphertexts with a specific access policy only. Such a decryption blackbox, which we call it a *policy-specific decryption blackbox*, has weaker decryption ability than that of the previous key-like decryption blackbox, as it decrypts ciphertexts with a specific access policy only rather than any arbitrary access policy as long as it is satisfied by a specific attribute set. In practice, a seller or a malicious user may set the price lower for such a policy-specific decryption blackbox, and advertises that it can decrypt any ciphertexts associated with access policy $\mathbb{A}_{\mathcal{D}}$. Below is another scenario, which we call it "found-in-the-wild", where policy-specific decryption blackboxes may be concerned.

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.

Though the corresponding attribute set is not available and only a specific access policy $\mathbb{A}_{\mathcal{D}}$ is known that the associated ciphertexts can be decrypted by a policy-specific decryption blackbox, interesting, we notice that the AugCP-ABE scheme in Sec. 4 also implies a fully secure BT-CP-ABE scheme with (selective) traceability against this policy-specific decryption blackbox. On its formal definition, it is similar to that of key-like decryption blackbox, with the following differences:

1. Trace algorithm: The tracing algorithm takes an access policy $\mathbb{A}_D$ as input, i.e. $\mathsf{Trace}^{\mathcal{D}}(\mathsf{PP}, \mathbb{A}_{\mathcal{D}}, \epsilon)$ and the rest is the same as before.
2. $\mathsf{Game}_{\mathsf{TR}}$: In the (Policy-specific) Decryption Blackbox Generation phase, the adversary $\mathcal{A}$ outputs a decryption blackbox $\mathcal{D}$ associated with an access policy $\mathbb{A}_{\mathcal{D}}$ and a non-negligible probability ϵ. $\mathcal{A}$ wins $\mathsf{Game}_{\mathsf{TR}}$ if
   (a) $\Pr[\mathcal{D}(\mathsf{Encrypt}(\mathsf{PP}, M, \mathbb{A}_{\mathcal{D}})) = M] \geq \epsilon$, where the probability is taken over the random choices of message M and the random coins of $\mathcal{D}$.
   (b) $\mathbb{K}_T = \emptyset$, or $\mathbb{K}_T \not\subseteq \mathbb{K}_{\mathcal{D}}$,
   or $(\forall k_t \in \mathbb{K}_T, S_{k_t}\ does\ not\ satisfy\ \mathbb{A}_{\mathcal{D}}\ )$.
3. $\mathsf{Game}_{\mathsf{IH}}$: We do not need the concept of strictest access policy here, i.e., in the Challenge phase, $\mathcal{A}$ submits a message M and an access policy $\mathbb{A}^*$, and the challenger sends $CT \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, \mathbb{A}^*, \bar{k} + b)$ to $\mathcal{A}$. Here we have to define a weaker model of $\mathsf{Game}_{\mathsf{IH}}$, where $\mathcal{A}$ is required to declare $\mathbb{A}^*$ before seeing the public parameter, and the defined index-hiding property is referred to as *selective* index-hiding.
4. Similar to Theorem 1, we can show that an AugCP-ABE with message-hiding and (selective) index-hiding properties implies a secure and (selectively) traceable BT-CP-ABE against policy-specific decryption blackbox.

To prove the message-hiding and index-hiding properties of the AugCP-ABE scheme under the definition above for policy-specific decryption blackbox, we only need to modify a few proofing details of Lemma 1, and the proof idea is similar to the current one for key-like decryption blackbox. In the proof of Lemma 1 for key-like decryption blackbox, as the challenge ciphertext is generated using the strictest access policy of the challenge attribute set S^*, we can have a guess on a particular attribute $\bar{x}$ and consequently prove the index-hiding against adaptive adversaries. However, in the case of policy-specific decryption blackbox, the challenge ciphertext is generated using the challenge access policy $\mathbb{A}^*$, where it is hard to have a successful guess unless we consider only the selective adversaries. In summary, the resulting BT-CP-ABE scheme is fully secure and selectively traceable against policy-specific decryption blackbox.

6 Related Work

Sahai and Waters [22] introduced Attribute-Based Encryption (ABE) for addressing the fuzzy identity matching problem in IBE. Goyal et al. [11] later formalized the notions of CP-ABE and Key-Policy ABE (KP-ABE). KP-ABE systems available in the literature include [21,15,20,1], however, these systems do not address the traceability problem.

Katz and Schröder [14] introduced the notion of traceability in the context of predicate encryption [13], where they proposed a generic construction that adds traceability to any inner-product predicate encryption (IPE) scheme with the price of adding overhead linear in $\mathcal{K}$ (the number of users) to the original scheme. Note that although IPE (e.g., the most expressive schemes to date in [13]) implies IBE, BE and KP-ABE, it cannot efficiently implement the functions of expressive CP-ABE. The advances of our work is making are twofold in the sense that we add traceability (1) to an existing expressive CP-ABE scheme (2) at the expense of sub-linear (i.e. $\sqrt{\mathcal{K}}$) overhead, although our result is specific rather than generic as [14] is.

7 Conclusion

In this paper, we proposed a new CP-ABE scheme that simultaneously supports fully collusion-resistant (and public) blackbox traceability and high expressivity (i.e., supporting any monotonic access structures), as well as without the one-use restriction. The scheme is proved secure against adaptive adversaries in the standard model. For the traceability against key-like decryption blackbox, the scheme is proved traceable against adaptive adversaries in the standard model, and for the traceability against policy-specific decryption blackbox, the scheme can be proved traceable against selective adversaries in the standard model. Compared with the most efficient conventional (non-traceable) CP-ABE schemes currently available with high expressivity and full security in the standard model, the new CP-ABE adds fully collusion-resistant (and public) blackbox traceability with the price of adding only $O(\sqrt{\mathcal{K}})$ elements in the ciphertext and public key.

8 Acknowledgments

The work was supported by the National Natural Science Foundation of China (No. 61033014 and No. 61161140320) and a grant from the RGC of the HKSAR, China (Project No. CityU 123511).

References

1. Attrapadung, N., Libert, B., de Panafieu, E.: Expressive key-policy attribute-based encryption with constant-size ciphertexts. In: Public Key Cryptography. Lecture Notes in Computer Science, vol. 6571, pp. 90–108. Springer (2011)
2. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy. pp. 321–334. IEEE Computer Society (2007)
3. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. In: CRYPTO. Lecture Notes in Computer Science, vol. 2139, pp. 213–229. Springer (2001)
4. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: CRYPTO. Lecture Notes in Computer Science, vol. 3621, pp. 258–275. Springer (2005)
5. Boneh, D., Sahai, A., Waters, B.: Fully collusion resistant traitor tracing with short ciphertexts and private keys. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 4004, pp. 573–592. Springer (2006)
6. Boneh, D., Waters, B.: A fully collusion resistant broadcast, trace, and revoke system. In: ACM Conference on Computer and Communications Security. pp. 211–220. ACM (2006)
7. Cheung, L., Newport, C.C.: Provably secure ciphertext policy abe. In: ACM Conference on Computer and Communications Security. pp. 456–465. ACM (2007)
8. Fiat, A., Naor, M.: Broadcast encryption. In: CRYPTO. Lecture Notes in Computer Science, vol. 773, pp. 480–491. Springer (1993)
9. Garg, S., Kumarasubramanian, A., Sahai, A., Waters, B.: Building efficient fully collusion-resilient traitor tracing and revocation schemes. In: ACM Conference on Computer and Communications Security. pp. 121–130. ACM (2010)
10. Goyal, V., Jain, A., Pandey, O., Sahai, A.: Bounded ciphertext policy attribute based encryption. In: ICALP (2). Lecture Notes in Computer Science, vol. 5126, pp. 579–591. Springer (2008)
11. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM Conference on Computer and Communications Security. pp. 89–98. ACM (2006)
12. Herranz, J., Laguillaumie, F., Ràfols, C.: Constant size ciphertexts in threshold attribute-based encryption. In: Public Key Cryptography. Lecture Notes in Computer Science, vol. 6056, pp. 19–34. Springer (2010)
13. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 4965, pp. 146–162. Springer (2008)
14. Katz, J., Schröder, D.: Tracing insider attacks in the context of predicate encryption schemes. In: ACITA (2011), `https://www.usukita.org/node/1779`

15. Lewko, A.B., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 6110, pp. 62–91. Springer (2010)
16. Lewko, A.B., Waters, B.: New proof methods for attribute-based encryption: Achieving full security through selective techniques. In: CRYPTO. Lecture Notes in Computer Science, vol. 7417, pp. 180–198. Springer (2012)
17. Li, J., Huang, Q., Chen, X., Chow, S.S.M., Wong, D.S., Xie, D.: Multi-authority ciphertext-policy attribute-based encryption with accountability. In: ASIACCS. pp. 386–390. ACM (2011)
18. Li, J., Ren, K., Kim, K.: A2be: Accountable attribute-based encryption for abuse free access control. IACR Cryptology ePrint Archive 2009, 118 (2009)
19. Liu, Z., Cao, Z., Wong, D.S.: White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures. IEEE Transactions on Information Forensics and Security 8(1), 76–88 (2013)
20. Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: CRYPTO. Lecture Notes in Computer Science, vol. 6223, pp. 191–208. Springer (2010)
21. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: ACM Conference on Computer and Communications Security. pp. 195–203. ACM (2007)
22. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 3494, pp. 457–473. Springer (2005)
23. Shamir, A.: Identity-based cryptosystems and signature schemes. In: CRYPTO. Lecture Notes in Computer Science, vol. 196, pp. 47–53. Springer (1984)
24. Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: Public Key Cryptography. Lecture Notes in Computer Science, vol. 6571, pp. 53–70. Springer (2011)

A Correctness

Correctness. Assume the encrypted message is M and the encryption index is $(\bar{i}, \bar{j})$. For $i \geq \bar{i}$ we have

$$\begin{aligned}
&\frac{e(K_{i,j}, Q_i) \cdot e(K'''_{i,j}, Q'''_i)}{e(K'_{i,j}, Q'_i) \cdot e(K''_{i,j}, Q''_i)} \\
=&\frac{e(g^{\alpha_i} g^{r_i c_j} f^{\sigma_{i,j}} h^{\delta_{i,j}} R, g^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}) e(Z_i^{\sigma_{i,j}} R''', g^{t_i})}{e(g^{\sigma_{i,j}} R', f^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)} Z_i^{t_i} f^{\pi}) e(g^{\delta_{i,j}} R'', h^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)})} \\
=&\frac{e(g^{\alpha_i}, g^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}) e(g^{r_i c_j}, g^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)})}{e(g^{\sigma_{i,j}}, f^{\pi})}.
\end{aligned}$$

If $i \geq \bar{i} \wedge j \geq \bar{j}$: we have

$$\begin{aligned}
&\frac{e_3(\boldsymbol{R}'_i, \boldsymbol{C}'_j)}{e_3(\boldsymbol{R}_i, \boldsymbol{C}_j)} = \frac{e_3(G_i^{\kappa s_i \boldsymbol{v}_i}, g^{\boldsymbol{w}_j})}{e_3(G_i^{s_i \boldsymbol{v}_i}, H_j^{\tau \boldsymbol{v}_c} \cdot g^{\kappa \boldsymbol{w}_j})} = \frac{1}{e_3(g^{r_i s_i \boldsymbol{v}_i}, g^{c_j \tau \boldsymbol{v}_c})} \\
&= \frac{1}{e(g,g)^{r_i s_i c_j \tau (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}},
\end{aligned}$$

If $i > \bar{i} \wedge j < \bar{j}$: note that for $i > \bar{i}$, we have $(\boldsymbol{v}_i \cdot \boldsymbol{\chi}_3) = 0$ (since $\boldsymbol{v}_i \in span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$), then we have

$$\begin{aligned}
&\frac{e_3(\boldsymbol{R}'_i, \boldsymbol{C}'_j)}{e_3(\boldsymbol{R}_i, \boldsymbol{C}_j)} = \frac{e_3(G_i^{\kappa s_i \boldsymbol{v}_i}, g^{\boldsymbol{w}_j})}{e_3(G_i^{s_i \boldsymbol{v}_i}, H_j^{\tau(\boldsymbol{v}_c + \mu_j \boldsymbol{\chi}_3)} \cdot g^{\kappa \boldsymbol{w}_j})} \\
&= \frac{1}{e_3(g^{r_i s_i \boldsymbol{v}_i}, g^{c_j \tau (\boldsymbol{v}_c + \mu_j \boldsymbol{\chi}_3)})} = \frac{1}{e(g,g)^{r_i s_i c_j \tau (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}},
\end{aligned}$$

If $i = \bar{i} \wedge j < \bar{j}$: note that for $i = \bar{i}$, we have that $(\boldsymbol{v}_i \cdot \boldsymbol{\chi}_3) \neq 0$ happens with overwhelming probability (since $\boldsymbol{v}_i$ is randomly chosen from $\mathbb{Z}_N^3$), then we have

$$\begin{aligned}\frac{e_3(\boldsymbol{R}'_i, \boldsymbol{C}'_j)}{e_3(\boldsymbol{R}_i, \boldsymbol{C}_j)} &= \frac{e_3(G_i^{\kappa s_i \boldsymbol{v}_i}, g^{\boldsymbol{w}_j})}{e_3(G_i^{s_i \boldsymbol{v}_i}, H_j^{\tau(\boldsymbol{v}_c + \mu_j \boldsymbol{\chi}_3)} \cdot g^{\kappa \boldsymbol{w}_j})} \\ &= \frac{1}{e_3(g^{r_i s_i \boldsymbol{v}_i}, g^{c_j \tau(\boldsymbol{v}_c + \mu_j \boldsymbol{\chi}_3)})} = \frac{1}{e(g,g)^{r_i s_i c_j \tau((\boldsymbol{v}_i \cdot \boldsymbol{v}_c) + \mu_j (\boldsymbol{v}_i \cdot \boldsymbol{\chi}_3))}},\end{aligned}$$

Thus from the values of T_i, D_P and D_I, for $M' = T_i/(D_P \cdot D_I)$ we have that: (1) if $(i > \bar{i}) \vee (i = \bar{i} \wedge j \geq \bar{j})$, then $M' = M$; (2) if $i = \bar{i} \wedge j < \bar{j}$, then $M' = M \cdot e(g,g)^{\tau s_i r_i c_j \mu_j (\boldsymbol{v}_i \cdot \boldsymbol{\chi}_3)}$; (3) if $i < \bar{i}$, then M' has no relation with M.

B Proofs

B.1 Proof of Theorem 2

Theorem 2 follows from the following Lemma 3 and Lemma 4.

Lemma 3. *[16] Under the 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, the CP-ABE scheme in [16] is fully secure.*

Proof. It follows from Theorem 1 of [16] immediately.

Lemma 4. *Suppose the CP-ABE scheme in [16] is fully secure. Then for our AugCP-ABE scheme no polynomial time adversary can win* $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$ *with non-negligible advantage.*

Proof. Suppose there is a PPT adversary $\mathcal{A}$ that can break our AugCP-ABE scheme Σ_{A} in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$ with non-negligible advantage $\mathsf{MH}_1^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}$, we construct a PPT algorithm $\mathcal{B}$ to break the CP-ABE scheme (denoted by Σ_{cpabe}) in [16] with advantage $Adv_{\mathcal{B}}\Sigma_{\mathsf{cpabe}}$, which equals to $\mathsf{MH}_1^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}$.

Setup. $\mathcal{B}$ receives the public parameter[6] $\mathsf{PP}^{\mathsf{cpabe}} = (N, g_3,\ g, g^a, g^b, e(g,g)^\alpha,\ \{U_x = g^{a_x}\}_{x \in \mathcal{U}})$ from the challenger, where $g \in \mathbb{G}_{p_1}$ and $g_3 \in \mathbb{G}_{p_3}$ are the generators of subgroups $\mathbb{G}_{p_1}$ and $\mathbb{G}_{p_3}$ respectively, and $a, b, \alpha, a_x (x \in \mathcal{U}) \in \mathbb{Z}_N$ are random exponents. $\mathcal{B}$ randomly chooses $\{\alpha'_i,\ r_i,\ z_i \in \mathbb{Z}_N\}_{i \in [m]}$, $\{c_j \in \mathbb{Z}_N\}_{j \in [m]}$, then gives $\mathcal{A}$ the public parameter PP:

$$\begin{aligned}&g,\ f = g^a,\ h = g^b,\ \ \{E_i = e(g,g)^\alpha e(g,g)^{\alpha'_i}\}_{i \in [m]}, \\ &\{G_i = g^{r_i},\ Z_i = g^{z_i}\}_{i \in [m]},\ \ \{H_j = g^{c_j}\}_{j \in [m]},\ \ \{U_x\}_{x \in \mathcal{U}}.\end{aligned}$$

Note that $\mathcal{B}$ implicitly chooses $\{\alpha_i \in \mathbb{Z}_N\}_{i \in [m]}$ such that $\{\alpha + \alpha'_i \equiv \alpha_i \bmod p_1\}_{i \in [m]}$.

[6] Note that: (1) we slightly changed the variable names in the underlying CP-ABE scheme to better suit our proof; and (2) in the original scheme of [16] g_3 is in the master secret key rather than in the public parameter, as g_3 is never used in encryption or decryption. Publishing g_3 in the public parameter will not affect the security of the scheme, as in the proof the simulator receives g_3 explicitly from the underlying assumptions and can provide it to the adversary in the public parameter.

Phase 1. To respond to $\mathcal{A}$'s query for $((i,j), S_{(i,j)})$, $\mathcal{B}$ submits $S_{(i,j)}$ to the challenger, and receives a private key $\mathsf{SK}^{\mathsf{cpabe}}_{S_{(i,j)}} = \big(\tilde{K} = g^{\alpha} g^{a\sigma} g^{b\delta} R,\ \tilde{K}' = g^{\sigma} R',\ \tilde{K}''_{i,j} = g^{\delta} R'',\ \ \{\tilde{K}_x = U_x^{\sigma} R_x\}_{x \in S_{(i,j)}}\big)$, where $\sigma, \delta \in \mathbb{Z}_N, R, R', R'', R_x(x \in S_{(i,j)}) \in \mathbb{G}_{p_3}$ are randomly chosen and unknown to $\mathcal{B}$. $\mathcal{B}$ randomly chooses $R''' \in \mathbb{G}_{p_3}$, then gives $\mathcal{A}$

$$\begin{aligned}\mathsf{SK}_{(i,j),S_{(i,j)}} &= \big(K_{i,j},\ K'_{i,j},\ K''_{i,j},\ K'''_{i,j},\ \{K_{i,j,x}\}_{x \in S_{(i,j)}}\big)\\ &= \big(\tilde{K} g^{\alpha'_i} g^{r_i c_j},\ \tilde{K}',\ \tilde{K}'',\ (\tilde{K}')^{z_i} R''',\ \{\tilde{K}_x\}_{x \in S_{(i,j)}}\big).\end{aligned}$$

Note that R''' makes the $\mathbb{G}_{p_3}$ part of $K'''_{i,j}$ uncorrelated to the $\mathbb{G}_{p_3}$ part of $K'_{i,j}$, this is why our simulator needs g_3. The distribution of the private keys is same with that of the real scheme, where $\sigma_{i,j}$ and $\delta_{i,j}$ are implicitly chosen such that $\sigma_{i,j} = \sigma, \delta_{i,j} = \delta$.

Challenge. $\mathcal{A}$ submits to $\mathcal{B}$ an LSSS matrix (A^*, ρ) and two equal length messages M_0, M_1. $\mathcal{B}$ submits $((A^*, \rho),\ M_0, M_1)$ to the challenger, and receives the challenge ciphertext in the form of $CT^{\mathsf{cpabe}} =$

$$\begin{aligned}\langle\ (A^*, \rho),\ \tilde{C} = M_b \cdot e(g,g)^{\alpha\tilde{\pi}},\ \tilde{C}_0 = g^{\tilde{\pi}},\ \tilde{C}'_0 = g^{b\tilde{\pi}},&\\ \{\tilde{C}_k = g^{aA^*_k \cdot \tilde{\boldsymbol{u}}} U_{\rho(k)}^{-\tilde{\xi}_k},\ \tilde{C}'_k = g^{\tilde{\xi}_k}\}_{k=1}^{l}\ \rangle,&\end{aligned}$$

where $\tilde{\boldsymbol{u}} = (\tilde{\pi}, \tilde{u}_2, \dots, \tilde{u}_n) \in \mathbb{Z}_N^n$ and $\{\tilde{\xi}_k \in \mathbb{Z}_N\}_{k=1}^{l}$ are randomly chosen and unknown to $\mathcal{B}$.

$\mathcal{B}$ randomly chooses $\kappa,\ \tau,\ s'_1, \dots, s'_m,\ t_1, \dots, t_m \in \mathbb{Z}_N,\ \boldsymbol{v}_c,\ \boldsymbol{w}_1, \dots, \boldsymbol{w}_m \in \mathbb{Z}_N^3,\ \xi'_1, \dots, \xi'_l \in \mathbb{Z}_N,\ \ \boldsymbol{u}' = (\pi', u'_2, \dots, u'_n) \in \mathbb{Z}_N^n$, and $r_x, r_y, r_z \in \mathbb{Z}_N$. Then it sets $\boldsymbol{\chi}_1 = (r_x, 0, r_z),\ \boldsymbol{\chi}_2 = (0, r_y, r_z), \boldsymbol{\chi}_3 = (-r_y r_z, -r_x r_z, r_x r_y)$, and creates the challenge ciphertext $CT = \langle (A^* \rho),\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, Q_i, Q'_i, Q''_i, Q'''_i,\ T_i)_{i=1}^{m},\ (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^{m},\ (P_k, P'_k)_{k=1}^{l} \rangle$ for $(\bar{i} = 1, \bar{j} = 1)$ as follows:

1. For each $i \in [m]$: since $\bar{i} = 1$, it randomly chooses $\boldsymbol{v}_1 \in \mathbb{Z}_N^3$ and $\boldsymbol{v}_i \in span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$ for $i > 1$. It sets
$$\begin{aligned}&\boldsymbol{R}_i = G_i^{s'_i \boldsymbol{v}_i} \cdot \tilde{C}_0^{\frac{r_i}{\tau(\boldsymbol{v}_i \cdot \boldsymbol{v}_c)} \boldsymbol{v}_i},\quad \boldsymbol{R}'_i = G_i^{\kappa s'_i \boldsymbol{v}_i} \cdot \tilde{C}_0^{\frac{\kappa r_i}{\tau(\boldsymbol{v}_i \cdot \boldsymbol{v}_c)} \boldsymbol{v}_i},\\ &Q_i = g^{\tau s'_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)} \tilde{C}_0,\quad Q'_i = f^{\tau s'_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)} Z_i^{t_i} f^{\pi'},\\ &Q''_i = h^{\tau s'_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)} \tilde{C}'_0,\quad Q'''_i = g^{t_i},\\ &T_i = \tilde{C} \cdot e(g^{\alpha'_i}, \tilde{C}_0) \cdot E_i^{\tau s'_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}.\end{aligned}$$
2. For each $j \in [m]$: $\boldsymbol{C}_j = H_j^{\tau \boldsymbol{v}_c} \cdot g^{\kappa \boldsymbol{w}_j},\ \ \boldsymbol{C}'_j = g^{\boldsymbol{w}_j}$.
3. For each $k \in [l]$: $P_k = f^{A^*_k \cdot \boldsymbol{u}'} U_{\rho(k)}^{-\xi'_k} / \tilde{C}_k,\ \ P'_k = g^{\xi'_k} / \tilde{C}'_k$.

Note that $\mathcal{B}$ implicitly chooses $s_1, \dots, s_m,\ \ \xi_1, \dots, \xi_l \in \mathbb{Z}_N$ and $\boldsymbol{u} = (\pi, u_2, \dots, u_n) \in \mathbb{Z}_N^n$ such that

$$\begin{aligned}s'_i + \frac{\tilde{\pi}}{\tau(\boldsymbol{v}_i \cdot \boldsymbol{v}_c)} \equiv s_i \bmod p_1\ \forall i \in \{1, \dots, m\},&\\ \xi'_k - \tilde{\xi}_k \equiv \xi_k \bmod p_1\ \forall k \in \{1, \dots, l\},&\\ \pi' - \tilde{\pi} \equiv \pi \bmod p_1,\ \ u'_d - \tilde{u}_d \equiv u_d \bmod p_1\ \forall d \in \{2, \dots, n\}.&\end{aligned}$$

Phase 2. Same with Phase 1.

Guess. $\mathcal{A}$ gives $\mathcal{B}$ a b'. $\mathcal{B}$ gives b' to the challenger.

Note that the distributions of the public parameter, private keys and challenge ciphertext that $\mathcal{B}$ gives $\mathcal{A}$ are same as the real scheme, we have $Adv_{\mathcal{B}} \Sigma_{\mathsf{cpabe}} = \mathsf{MH}_1^{\mathsf{A}} \mathsf{Adv}_{\mathcal{A}}$.

B.2 Proof of Lemma 1

Proof. Suppose there exists a polynomial time adversary $\mathcal{A}$ that breaks the index-hiding game with advantage ϵ. We build a PPT algorithm $\mathcal{B}$ to solve a 3-Party Diffie-Hellman problem instance in a subgroup as follows.

$\mathcal{B}$ receives the 3-Party Diffie-Hellman challenge in a subgroup from the challenger as $(N, \mathbb{G}, \mathbb{G}_T, e, g, g_2, g_3,\ A = g^a, B = g^b, C = g^c, T)$, where $\mathbb{G}$ is a bilinear group of order $N = p_1 p_2 p_3$, $\mathbb{G}_{p_i}$ is the subgroup of order p_i in $\mathbb{G}$ $(i = 1, 2, 3)$, g, g_2 and g_3 are generators of $\mathbb{G}_{p_1}$, $\mathbb{G}_{p_2}$ and $\mathbb{G}_{p_3}$ respectively, and a, b, c are randomly chosen from $\mathbb{Z}_N$. $\mathcal{B}$'s goal is to determine $T = g^{abc}$ or T is a random element from $\mathbb{G}_{p_1}$.

Setup. Firstly, $\mathcal{B}$ randomly chooses an attribute $\bar{x} \in \mathcal{U}$ to guess that $\bar{x}$ will be in the challenge attribute set S^* (regardless of whether $\mathcal{A}$ behaves in **Case I** or **Case II**) and will not be in $S_{(\bar{i},\bar{j})}$ if $\mathcal{A}$ behaves in **Case II**. Then $\mathcal{B}$ randomly chooses

$$\{\alpha_i \in \mathbb{Z}_N\}_{i\in[m]},\ \{r_i,\ z'_i \in \mathbb{Z}_N\}_{i\in[m]\setminus\{\bar{i}\}},\ \{c_j \in \mathbb{Z}_N\}_{j\in[m]\setminus\{\bar{j}\}},$$
$$\{a_x \in \mathbb{Z}_N\}_{x\in\mathcal{U}\setminus\{\bar{x}\}},\ \ r'_{\bar{i}},\ z_{\bar{i}},\ c'_{\bar{j}},\ a'_{\bar{x}} \in \mathbb{Z}_N,$$

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

$$\Big(g,\ f = C^{\eta},\ h = g^{\theta},\ \ \{E_i = e(g,g)^{\alpha_i}\}_{i\in[m]},$$
$$\{G_i = g^{r_i},\ Z_i = C^{z'_i}\}_{i\in[m]\setminus\{\bar{i}\}},\ G_{\bar{i}} = B^{r'_{\bar{i}}},\ Z_{\bar{i}} = g^{z_{\bar{i}}}$$
$$\{H_j = g^{c_j}\}_{j\in[m]\setminus\{\bar{j}\}},\ H_{\bar{j}} = C^{c'_{\bar{j}}},$$
$$\{U_x = g^{a_x}\}_{x\in\mathcal{U}\setminus\{\bar{x}\}},\ U_{\bar{x}} = C^{a'_{\bar{x}}}\ \Big).$$

Note that $\mathcal{B}$ implicitly chooses $r_{\bar{i}}, c_{\bar{j}}, a_{\bar{x}} \in \mathbb{Z}_N$ and $\{z_i \in \mathbb{Z}_N\}_{i\in[m]\setminus\{\bar{i}\}}$ such that

$$br'_{\bar{i}} \equiv r_{\bar{i}} \bmod p_1,\quad cc'_{\bar{j}} \equiv c_{\bar{j}} \bmod p_1,\quad ca'_{\bar{x}} \equiv a_{\bar{x}} \bmod p_1,$$
$$cz'_i \equiv z_i \bmod p_1\ \forall i \in [m] \setminus \{\bar{i}\}.$$

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 $\mathsf{SK}_{(i,j),S_{(i,j)}} = \big(K_{i,j},\ K'_{i,j},\ K''_{i,j},\ K'''_{i,j},\ \ \{K_{i,j,x}\}_{x\in S_{(i,j)}}\big)$ as

$$K_{i,j} = \begin{cases} g^{\alpha_i} g^{r_i c_j} f^{\sigma_{i,j}} h^{\delta_{i,j}} R, & : i \neq \bar{i}, j \neq \bar{j} \\ g^{\alpha_i} B^{r'_i c_j} f^{\sigma_{i,j}} h^{\delta_{i,j}} R, & : i = \bar{i}, j \neq \bar{j} \\ g^{\alpha_i} C^{r_i c'_j} f^{\sigma_{i,j}} h^{\delta_{i,j}} R, & : i \neq \bar{i}, j = \bar{j} \end{cases}$$
$$K'_{i,j} = g^{\sigma_{i,j}} R',\ K''_{i,j} = g^{\delta_{i,j}} R'',\ K'''_{i,j} = Z_i^{\sigma_{i,j}} R''',$$
$$K_{i,j,x} = U_x^{\sigma_{i,j}} R_x\ \forall x \in S_{(i,j)}.$$

- if $(i,j) = (\bar{i},\bar{j})$: it means that $\mathcal{A}$ behaves in **Case II**. If $\bar{x} \in S_{(i,j)}$, then $\mathcal{B}$ aborts and outputs a random $b' \in \{0,1\}$ to the challenger. Otherwise, $\mathcal{B}$ randomly chooses $\sigma'_{\bar{i},\bar{j}} \in \mathbb{Z}_N$ and sets the value of $\sigma_{\bar{i},\bar{j}}$ by implicitly setting $\sigma'_{\bar{i},\bar{j}} - br'_{\bar{i}}c'_{\bar{j}}/\eta \equiv \sigma_{\bar{i},\bar{j}} \bmod p_1$. 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 $\mathsf{SK}_{(\bar{i},\bar{j}),S_{(\bar{i},\bar{j})}} = \big(K_{\bar{i},\bar{j}},\ K'_{\bar{i},\bar{j}},\ K''_{\bar{i},\bar{j}},\ K'''_{\bar{i},\bar{j}},\ \{K_{\bar{i},\bar{j},x}\}_{x \in S_{(\bar{i},\bar{j})}}\big)$ as

$$K_{\bar{i},\bar{j}} = g^{\alpha_{\bar{i}}} f^{\sigma'_{\bar{i},\bar{j}}} h^{\delta_{\bar{i},\bar{j}}} R, \quad K'_{\bar{i},\bar{j}} = g^{\sigma'_{\bar{i},\bar{j}}} B^{-r'_{\bar{i}} c'_{\bar{j}}/\eta} R',$$
$$K''_{\bar{i},\bar{j}} = g^{\delta_{\bar{i},\bar{j}}} R'', \quad K'''_{\bar{i},\bar{j}} = (g^{\sigma'_{\bar{i},\bar{j}}} B^{-r'_{\bar{i}} c'_{\bar{j}}/\eta})^{z_{\bar{i}}} R''',$$
$$K_{\bar{i},\bar{j},x} = (g^{\sigma'_{\bar{i},\bar{j}}} B^{-r'_{\bar{i}} c'_{\bar{j}}/\eta})^{a_x} R_x \ \forall x \in S_{(i,j)}.$$

Challenge. $\mathcal{A}$ submits a message M and an attribute set S^*. If $\bar{x} \notin S^*$ then $\mathcal{B}$ aborts and outputs a random $b' \in \{0,1\}$ to the challenger. Otherwise, $\mathcal{B}$ constructs the LSSS matrix (A^*, ρ) for $\mathbb{A}_{S^*}$. Let $l \times n$ be the size of (A^*, ρ). Note that $S^* \setminus \{\bar{x}\}$ does not satisfy $\mathbb{A}_{S^*}$, $\mathcal{B}$ first computes a vector $\bar{\boldsymbol{u}} \in \mathbb{Z}_N^n$ that has first entry equal to 1 and is orthogonal to all of the rows A^*_k of A^* such that $\rho(k) \in S^* \setminus \{\bar{x}\}$ (such a vector must exist since $S^* \setminus \{\bar{x}\}$ fails to satisfy (A^*, ρ), and it is efficiently computable). $\mathcal{B}$ randomly chooses

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

with the first entry of $\boldsymbol{u}'$ equal to zero. It also randomly chooses $r_x, r_y, r_z \in \mathbb{Z}_N$, and sets $\boldsymbol{\chi}_1 = (r_x, 0, r_z), \boldsymbol{\chi}_2 = (0, r_y, r_z), \boldsymbol{\chi}_3 = \boldsymbol{\chi}_1 \times \boldsymbol{\chi}_2 = (-r_y r_z, -r_x r_z, r_x r_y)$.

$\mathcal{B}$ randomly chooses $(\nu_{c,1}, \nu_{c,2}, \nu_{c,3}) \in \mathbb{Z}_N^3$. Let $\boldsymbol{v}_c^p = \nu_{c,1}\boldsymbol{\chi}_1 + \nu_{c,2}\boldsymbol{\chi}_2$ and $\boldsymbol{v}_c^q = \nu_{c,3}\boldsymbol{\chi}_3$, implicitly setting $\boldsymbol{v}_c = a^{-1}\boldsymbol{v}_c^p + \boldsymbol{v}_c^q$, $\mathcal{B}$ creates the ciphertext $\langle (A, \rho),\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, Q_i, Q'_i, Q''_i, Q'''_i, T_i)_{i=1}^m,\ (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^m,\ (P_k, P'_k)_{k=1}^l \rangle$ as follows:

1. For each row $i \in [m]$:
   - if $i < \bar{i}$: it randomly chooses $\boldsymbol{v}_i \in \mathbb{Z}_N^3$ and $\hat{s}_i \in \mathbb{Z}_N$, then sets

$$\boldsymbol{R}_i = g^{\boldsymbol{v}_i}, \ \boldsymbol{R}'_i = B^{\boldsymbol{v}_i}, \ \ Q_i = g^{s_i}, \ Q'_i = f^{s_i} Z_i^{t'_i} f^{\pi'},$$
$$Q''_i = h^{s_i}, \ Q'''_i = g^{t'_i} A^{\eta\tau' s'_{\bar{i}} (\boldsymbol{v}_{\bar{i}} \cdot \boldsymbol{v}_c^q)/z'_i}, \ \ T_i = E_i^{\hat{s}_i}.$$

   - if $i = \bar{i}$: it randomly chooses $\boldsymbol{v}_{\bar{i}} \in \mathbb{Z}_N^3$, then sets

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

   - if $i > \bar{i}$: it randomly chooses $\boldsymbol{v}_i \in \mathit{span}\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$, then sets

$$\begin{aligned} &\boldsymbol{R}_i = g^{r_i s_i \boldsymbol{v}_i}, \ \ \boldsymbol{R}'_i = B^{r_i s_i \boldsymbol{v}_i}, \\ &Q_i = B^{\tau' s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c^p)}, \ Q'_i = Z_i^{t'_i} f^{\pi'}, \ Q''_i = Q_i^{\theta}, \ Q'''_i = g^{t'_i} B^{\frac{-\eta\tau' s'_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c^p)}{z'_i}} A^{\frac{\eta\tau' s'_{\bar{i}} (\boldsymbol{v}_{\bar{i}} \cdot \boldsymbol{v}_c^q)}{z'_i}}, \\ &T_i = M \cdot e(g^{\alpha_i}, Q_i). \end{aligned}$$

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 $(ab)^{-1}\mu'_j\nu_{c,3} - \nu_{c,3} \equiv \mu_j \bmod N$, then sets

$$\boldsymbol{C}_j = B^{c_j\tau'\boldsymbol{v}_c^p} \cdot g^{c_j\tau'\mu'_j\boldsymbol{v}_c^q} \cdot B^{\boldsymbol{w}_j}, \quad \boldsymbol{C}'_j = g^{\boldsymbol{w}_j}.$$

   - if $j = \bar{j}$: $\boldsymbol{C}_j = T^{c'_{\bar{j}}\tau'\boldsymbol{v}_c^q} \cdot B^{\boldsymbol{w}'_j}, \quad \boldsymbol{C}'_j = g^{\boldsymbol{w}'_{\bar{j}}} \cdot C^{-c'_{\bar{j}}\tau'\boldsymbol{v}_c^p}$.
   - if $j > \bar{j}$: $\boldsymbol{C}_j = B^{c_j\tau'\boldsymbol{v}_c^p} \cdot B^{\boldsymbol{w}'_j}, \quad \boldsymbol{C}'_j = g^{\boldsymbol{w}'_j} \cdot A^{-c_j\tau'\boldsymbol{v}_c^q}$.
3. For each $k \in [l]$:
   - if $\rho(k) \neq \bar{x}$: it sets $P_k = f^{A_k^*\cdot\boldsymbol{u}'}U_{\rho(k)}^{-\xi_k}, \quad P'_k = g^{\xi_k}$.
   - if $\rho(k) = \bar{x}$: it sets

$$\begin{aligned} P_k &= f^{\pi' A_k^*\cdot\bar{\boldsymbol{u}}} f^{A_k^*\cdot\boldsymbol{u}'} C^{-a_{\bar{x}}\xi'_k}, \\ P'_k &= g^{\xi'_k} A^{-\eta\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}\cdot\boldsymbol{v}_c^q)(A_k^*\cdot\bar{\boldsymbol{u}})/a_{\bar{x}}}. \end{aligned}$$

Note that $\mathcal{B}$ implicitly chooses $\kappa, \tau,\ s_{\bar{i}},\ t_i(i \in [m] \setminus \{\bar{i}\}) \in \mathbb{Z}_N$, $\boldsymbol{w}_j \in \mathbb{Z}_N^3(j \in \{\bar{j}, \ldots, m\})$, $\pi \in \mathbb{Z}_N$, $\boldsymbol{u} \in \mathbb{Z}_N^n$, and $\{\xi_k \in \mathbb{Z}_N\}_{k\in[l]\ s.t.\ \rho(k)=\bar{x}}$ such that

$$\begin{aligned} &b \equiv \kappa \bmod p_1, \quad ab\tau' \equiv \tau \bmod p_1, \quad s'_{\bar{i}}/b \equiv s_{\bar{i}} \bmod p_1, \\ &\forall i \in \{1, \ldots, \bar{i}-1\} : \quad t'_i + \eta a\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}} \cdot \boldsymbol{v}_c^q)/z'_i \equiv t_i \bmod p_1, \\ &\forall i \in \{\bar{i}+1, \ldots, m\} : \\ &\quad t'_i - \eta b\tau' s'_i(\boldsymbol{v}_i \cdot \boldsymbol{v}_c^p)/z'_i + \eta a\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}} \cdot \boldsymbol{v}_c^q)/z'_i \equiv t_i \bmod p_1, \\ &\quad \boldsymbol{w}'_{\bar{j}} - cc'_{\bar{j}}\tau'\boldsymbol{v}_c^p \equiv \boldsymbol{w}_{\bar{j}} \bmod p_1, \\ &\forall j \in \{\bar{j}+1, \ldots, m\} : \quad \boldsymbol{w}'_j - ac_j\tau'\boldsymbol{v}_c^q \equiv \boldsymbol{w}_j \bmod p_1, \\ &\quad \pi' - a\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}} \cdot \boldsymbol{v}_c^q) \equiv \pi \bmod p_1, \ \boldsymbol{u} = \pi\bar{\boldsymbol{u}} + \boldsymbol{u}', \\ &\forall k \in [l]\ s.t.\ \rho(k) = \bar{x} : \\ &\quad \xi'_k - \eta a\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}} \cdot \boldsymbol{v}_c^q)(A_k^* \cdot \bar{\boldsymbol{u}})/a_{\bar{x}} \equiv \xi_k \bmod p_1. \end{aligned}$$

If $T = g^{abc}$, then the ciphertext is a well-formed encryption to the index $(\bar{i}, \bar{j})$. If T is randomly chosen, say $T = g^r$ 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 $(\frac{r}{abc} - 1)\nu_{c,3} \equiv \mu_{\bar{j}} \bmod p_1$.

Guess. $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ to $\mathcal{B}$, then $\mathcal{B}$ outputs this b' to the challenger.

Note that when $\mathcal{B}$ does not abort, the distributions of the public parameter, private keys and challenge ciphertext are same as the real scheme. As $S^* \neq \emptyset$ and if $\mathcal{A}$ behaves in **Case II** then the attribute set $S_{(\bar{i},\bar{j})}$ must satisfy $S^* \setminus S_{(\bar{i},\bar{j})} \neq \emptyset$, the event that $\mathcal{B}$ does not abort will happen with probability at least $1/|\mathcal{U}|$. Thus, $\mathcal{B}$'s advantage in the 3-Party Diffie-Hellman game will be at least $\epsilon/|\mathcal{U}|$. As of the fully secure CP-ABE schemes in [15,20,16], the size of attribute universe (i.e. $|\mathcal{U}|$) in our scheme is also polynomial in the security parameter λ. Thus a degradation of $O(1/|\mathcal{U}|)$ in the security reduction is acceptable.

B.3 Proof of Claim 2

Garg et al. [9, Sec. 5.1] proposed an AugBE scheme $\varSigma_{\mathsf{AugBE}} = (\mathsf{Setup}_{\mathsf{AugBE}}, \mathsf{Encrypt}_{\mathsf{AugBE}}, \mathsf{Decrypt}_{\mathsf{AugBE}})$ and proved that $\varSigma_{\mathsf{AugBE}}$ has index-hiding property. In their proof of Lemma 6.3 in [9], two hybrid experiments

- H_2^{AugBE}: Encrypt to $(\bar{i}, m+1)$, (i.e. H_2 in [9])
- H_3^{AugBE}: Encrypt to $(\bar{i}+1, 1)$, (i.e. H_5 in [9])

were defined and proved indistinguishable by a sequence of hybrid sub-experiments. Our Claim 2 follows from the following Claim 3 and 4.

Claim 3. *[9] Suppose that the 3-Party Diffie-Hellman Assumption and the Decisional Linear Assumption hold. Then for scheme $\varSigma_{\mathsf{AugBE}}$ no polynomial time adversary can distinguish between experiments H_2^{AugBE} and H_3^{AugBE} with non-negligible advantage.*

Proof. This claim follows from the Lemma 6.3 in [9].

Claim 4. *Suppose that for scheme $\varSigma_{\mathsf{AugBE}}$ no polynomial time adversary can distinguish between experiments H_2^{AugBE} and H_3^{AugBE} with non-negligible advantage. Then for our AugCP-ABE scheme $\varSigma_{\mathsf{A}}$ no polynomial time adversary can distinguish between experiments H_2 and H_3 with non-negligible advantage.*

Proof. Suppose there is a PPT adversary $\mathcal{A}$ that can distinguish between H_2 and H_3 for our AugCP-ABE scheme $\varSigma_{\mathsf{A}}$ with non-negligible advantage, we construct a PPT algorithm $\mathcal{B}$ to distinguish between H_2^{AugBE} and H_3^{AugBE} for $\varSigma_{\mathsf{AugBE}}$ with non-negligible advantage.

The game of $\mathcal{B}$ distinguishing between H_2^{AugBE} and H_3^{AugBE} is played in the subgroup $\mathbb{G}_{p_1}$ of order p_1 in a composite order group $\mathbb{G}_N$ of order $N = p_1 p_2 p_3$. $\mathcal{B}$ is given the values of p_1, p_2 and p_3, and can chooses for itself everything in the subgroup $\mathbb{G}_{p_3}$.

Setup. The challenger gives $\mathcal{B}$ the public key $\mathsf{PK}^{\mathsf{AugBE}}$, and due to $(\bar{i}, m+1) \notin \{(i,j) | 1 \le i, j \le m\}$, the challenger gives $\mathcal{B}$ all private keys in the set $\{\mathsf{SK}_{(i,j)}^{\mathsf{AugBE}} | 1 \le i, j \le m\}$:[7]

$$\begin{aligned}\mathsf{PK}^{\mathsf{AugBE}} =& \big(\ g,\ \{E_i = e(g,g)^{\alpha_i},\ G_i = g^{r_i}\}_{i \in [m]},\\ & \quad \{H_j = g^{c_j},\ f_j\}_{j \in [m]}\ \big),\\ \mathsf{SK}_{(i,j)}^{\mathsf{AugBE}} =& \big(\tilde{K}_{i,j},\ \tilde{K}'_{i,j},\ \ \{\tilde{K}_{i,j,\tilde{j}}\}_{1 \le \tilde{j} \le m, \tilde{j} \ne j}\ \big)\\ =& \big(g^{\alpha_i} g^{r_i c_j} f_j^{\sigma_{i,j}},\ g^{\sigma_{i,j}},\ \ \{f_{\tilde{j}}^{\sigma_{i,j}}\}_{1 \le \tilde{j} \le m, \tilde{j} \ne j}\ \big),\end{aligned}$$

where generator $g \in \mathbb{G}_{p_1}$, elements $f_1, \dots, f_m \in \mathbb{G}_{p_1}$ and exponents $\{\alpha_i, r_i \in \mathbb{Z}_{p_1}\}_{i \in [m]}, \{c_j \in \mathbb{Z}_{p_1}\}_{j \in [m]}$, $\sigma_{i,j} (1 \le i, j \le m) \in \mathbb{Z}_{p_1}$ are randomly chosen.

$\mathcal{B}$ randomly chooses $\theta,\ z_1, \dots, z_m,\ a_x (x \in \mathcal{U}) \in \mathbb{Z}_N$, then gives $\mathcal{A}$ the following public parameter PP:

$$\begin{aligned}g,\ f = \prod_{1 \le j \le m} f_j,\ h = g^{\theta},\ \ \{E_i,\ G_i,\ Z_i = g^{z_i}\}_{i \in [m]},&\\ \{H_j\}_{j \in [m]},\ \ \{U_x = g^{a_x}\}_{x \in \mathcal{U}}.&\end{aligned}$$

[7] Note that we slightly changed the variable names in the underlying AugBE scheme to better suit our proof.

Key Query. To respond to $\mathcal{A}$'s query for $((i,j), S_{(i,j)})$, $\mathcal{B}$ randomly chooses $\delta_{i,j} \in \mathbb{Z}_N$ and R, R', R'', R''', $R_x (x \in S_{(i,j)}) \in \mathbb{G}_{p_3}$, and creates the private key $\mathsf{SK}_{(i,j),S_{(i,j)}} = \big(K_{i,j},\ K'_{i,j},\ K''_{i,j},\ K'''_{i,j},\ \{K_{i,j,x}\}_{x \in S_{(i,j)}} \big)$ from $\mathsf{SK}^{\mathsf{AugBE}}_{(i,j)}$ as

$$K_{i,j} = \tilde{K}_{i,j} \cdot \prod_{\tilde{j} \in [m] \setminus \{j\}} \tilde{K}_{i,j,\tilde{j}} \cdot h^{\delta_{i,j}} \cdot R,$$
$$K'_{i,j} = \tilde{K}'_{i,j} \cdot R', \quad K''_{i,j} = g^{\delta_{i,j}} \cdot R'', \quad K'''_{i,j} = (\tilde{K}'_{i,j})^{z_i} \cdot R''',$$
$$K_{i,j,x} = (\tilde{K}'_{i,j})^{a_x} \cdot R_x \ \forall x \in S_{(i,j)}.$$

Challenge. $\mathcal{A}$ submits a message M and an attribute set S^*. Note that $(\bar{i}, m+1) \notin \{(i,j) | 1 \le i,j \le m\}$, $\mathcal{B}$ sets $Y = \{(i,j) | 1 \le i,j \le m\}$ and submits (M, Y) to the challenger. The challenger gives $\mathcal{B}$ the challenge ciphertext $CT^{\mathsf{AugBE}} = \langle (\tilde{\boldsymbol{R}}_i, \tilde{\boldsymbol{R}}'_i, \tilde{Q}_i, \tilde{Q}'_i, \tilde{T}_i)_{i=1}^m,\ (\tilde{\boldsymbol{C}}_j, \tilde{\boldsymbol{C}}'_j)_{j=1}^m,\ Y \rangle$, which is encrypted to $(i^*, j^*) \in \{(\bar{i}, m+1), (\bar{i}+1, 1)\}$ and in the form of

1. For each $i \in [m]$:
   – if $i < i^*$: $\tilde{\boldsymbol{R}}_i = g^{\boldsymbol{v}_i}$, $\tilde{\boldsymbol{R}}'_i = g^{\kappa \boldsymbol{v}_i}$, $\tilde{Q}_i = g^{s_i}$,
   $\tilde{Q}'_i = (\prod_{\hat{j} \in Y_i} f_{\hat{j}})^{s_i}$, $\tilde{T}_i = E_i^{\hat{s}_i}$.
   – if $i \ge i^*$: $\tilde{\boldsymbol{R}}_i = G_i^{s_i \boldsymbol{v}_i}$, $\tilde{\boldsymbol{R}}'_i = G_i^{\kappa s_i \boldsymbol{v}_i}$, $\tilde{Q}_i = g^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}$,
   $\tilde{Q}'_i = (\prod_{\hat{j} \in Y_i} f_{\hat{j}})^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}$, $\tilde{T}_i = M \cdot E_i^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}$.
2. For each $j \in [m]$:
   – if $j < j^*$: $\tilde{\boldsymbol{C}}_j = H_j^{\tau(\boldsymbol{v}_c + \mu_j \boldsymbol{\chi}_3)} \cdot g^{\kappa \boldsymbol{w}_j}$, $\tilde{\boldsymbol{C}}'_j = g^{\boldsymbol{w}_j}$.
   – if $j \ge j^*$: $\tilde{\boldsymbol{C}}_j = H_j^{\tau \boldsymbol{v}_c} \cdot g^{\kappa \boldsymbol{w}_j}$, $\tilde{\boldsymbol{C}}'_j = g^{\boldsymbol{w}_j}$.

where $\kappa, \tau, s_i (1 \le i \le m), \hat{s}_i (1 \le i < i^*), \mu_j (1 \le j < j^*) \in \mathbb{Z}_{p_1}$, $\boldsymbol{v}_c, \boldsymbol{w}_j (1 \le j \le m), \boldsymbol{v}_i (1 \le i \le i^*) \in \mathbb{Z}_{p_1}^3$, and $\boldsymbol{v}_i (i > i^*) \in span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$ are randomly chosen (where $\boldsymbol{\chi}_1 = (r_x, 0, r_z), \boldsymbol{\chi}_2 = (0, r_y, r_z), \boldsymbol{\chi}_3 = (-r_y r_z, -r_x r_z, r_x r_y)$ are for randomly chosen $r_x, r_y, r_z \in \mathbb{Z}_{p_1}$), and $Y_i = \{j | (i,j) \in Y\}$.

Note that $Y = \{(i,j) | 1 \le i,j \le m\}$ so that $Y_i = \{1, \ldots, m\}$ for all $1 \le i \le m$, we have that $\tilde{Q}'_i = (\prod_{\hat{j} \in Y_i} f_{\hat{j}})^{s_i} = f^{s_i}$ for $i < i^*$ and $\tilde{Q}'_i = (\prod_{\hat{j} \in Y_i} f_{\hat{j}})^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)} = f^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}$ for $i \ge i^*$.

$\mathcal{B}$ constructs the LSSS matrix (A^*, ρ) for $\mathbb{A}_{S^*}$. Let A^* be an $l \times n$ matrix, $\mathcal{B}$ randomly chooses $t_1, \ldots, t_m, \xi_1, \ldots, \xi_l \in \mathbb{Z}_N$, $\boldsymbol{u} = (\pi, u_2, \ldots, u_n) \in \mathbb{Z}_N^n$, then creates the ciphertext $\langle (A, \rho),\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, Q_i, Q'_i, Q''_i, Q'''_i, T_i)_{i=1}^m,\ (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^m,\ (P_k, P'_k)_{k=1}^l \rangle$ as follows:

1. For each $i \in [m]$: $\boldsymbol{R}_i = \tilde{\boldsymbol{R}}_i$, $\boldsymbol{R}'_i = \tilde{\boldsymbol{R}}'_i$, $Q_i = \tilde{Q}_i$, $Q'_i = \tilde{Q}'_i \cdot Z_i^{t_i} f^{\pi}$, $Q''_i = Q_i^{\theta}$, $Q'''_i = g^{t_i}$, $T_i = \tilde{T}_i$.
2. For each $j \in [m]$: $\boldsymbol{C}_j = \tilde{\boldsymbol{C}}_j$, $\boldsymbol{C}'_j = \tilde{\boldsymbol{C}}'_j$.
3. For each $k \in [l]$: $P_k = f^{A_k^* \cdot \boldsymbol{u}} U_{\rho(k)}^{-\xi_k}$, $P'_k = g^{\xi_k}$.

Guess. $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ to $\mathcal{B}$, then $\mathcal{B}$ outputs this b' to the challenger as its answer to distinguish between H_2^{AugBE} and H_3^{AugBE} for scheme Σ_{AugBE}.

As the exponents are applied only to the elements in the subgroup $\mathbb{G}_{p_1}$, from the view of $\mathcal{A}$, the distributions of the public parameter, private keys and challenge ciphertext that $\mathcal{B}$ gives $\mathcal{A}$ are same as the real scheme. Thus $\mathcal{B}$'s advantage in distinguishing between H_2^{AugBE} and H_3^{AugBE} for scheme Σ_{AugBE} will be exactly equal to $\mathcal{A}$'s advantage in distinguishing between H_2 and H_3 for scheme Σ_{A}.