Expressive Black-box Traceable Ciphertext-Policy Attribute-Based Encryption

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

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

Abstract. In a Ciphertext-Policy Attribute-Based Encryption (CP-ABE) system, decryption privileges are defined over attributes that could be shared by multiple users. If some of the users leak their decryption privileges to the public or to some third party, say for profit gain, a conventional CP-ABE has no tracing mechanism for finding these malicious users out. There are two levels of traceability for tackling this problem: (1) given a well-formed decryption key, a *White-Box* tracing algorithm can find out the original key owner; and (2) given a decryption-device while the underlying decryption algorithm or key may not be given, a *Black-Box* tracing algorithm, which treats the decryption-device as an oracle, can find out at least one of the malicious users whose keys have been used for constructing the decryption-device. In this paper we propose the first *Expressive Black-box Traceable CP-ABE* system which has two main merits: (1) it supports fully collusion-resistant black-box traceability, that is, an adversary is allowed to access an arbitrary number of keys of its own choice when building the decryption-device, and (2) it is highly expressive, that is, the system supports policies expressed in any monotonic access structures. In addition, the traceability of this new system is public, that no secret input is required and no authority needs to be called in, instead, anyone can run the tracing algorithm. We show that the system is secure against adaptive adversaries in the standard model, and is efficient, that when compared with the expressive (non-traceable) CP-ABE due to Lewko et al. in Eurocrypt 2010, our new system *adds* fully collusion-resistant black-box traceability with the price of adding only $O(\sqrt{\mathcal{K}})$ elements into the ciphertext and public key, rather than increasing the sizes linearly with $\mathcal{K}$, which is the number of users in the system.

Keywords: Attribute-Based Encryption, Ciphertext-Policy, Black-Box Traceability

1 Introduction

Ciphertext-Policy Attribute-Based Encryption (CP-ABE), introduced by Goyal et al. [15], provides a new flexible and efficient mechanism for realizing one-to-many encryption, and due to its flexible expressiveness it is regarded as a promising tool for enforcing fine-grained access control over encrypted data. For example, suppose Alice wants to encrypt some messages to all PhD students and alumni in the Department of Computer Science, but she does not know or is not possible to find out the identities of all the eligible receivers. One way to think about this problem is to have Alice encrypt the messages with an access policy defined over some descriptive attributes such as "(COMPUTER SCIENCE **AND** (PhD STUDENT **OR** ALUMNI))", so that only receivers who have attributes satisfying this policy can decrypt. Traditional Public Key Encryption (PKE) and Identity-Based Encryption (IBE) [31,4] are designed for one-to-one encryption and therefore, will be very inefficient for solving this problem. Broadcast Encryption (BE) [12] is not suitable either, as in a BE system the encryptor must know and specify the exact identities/indices of the receivers. CP-ABE provides an elegant solution to the application above. In a CP-ABE system, each user is issued a decryption key by an authority according to the attributes he possesses. An encryptor decides what attributes an eligible receiver should have by encrypting a message with an access policy defined over some attributes. If and only if a user's attributes satisfy the access policy of a ciphertext, can he decrypt the ciphertext. Since the first CP-ABE scheme was proposed by Bethencourt, Sahai and Waters [3], a series of work [10,14,33,21,28,16] has been done to achieve better expressiveness, efficiency or security. In particular, the schemes by Lewko et al. [21] and by Okamoto and Takashima [28] are expressive, efficient and provably secure against adaptive adversaries in the standard model. And the most recent advances are duo to Lewko and Waters [22], where they proposed a new proof technique and obtained two new CP-ABE schemes that are also expressive, efficient and fully secure in the standard model. Additionally their new CP-ABE schemes eliminate the one-use restriction (i.e. a single attribute could only be used once in a policy) which the schemes in [21,28] suffer from[3].

However, there is a major issue that needs to be solved as it limits the applications of CP-ABE to date. The advantage of efficiency and policy expressiveness of CP-ABE comes from the fact that the access policies do not have to contain identity-related attributes such as identities or names, and are always built over role-based attributes, which implies that decryption keys with eligible attributes corresponding to the access policies generally do not have identity information embedded, i.e., those decryption keys are non-traceable by nature, as the attributes are generally shared by multiple users. For example, both Bob (with attributes {Bob, PhD, Computer Science}) and Tom (with attributes {Tom, PhD, Computer Science}) could share a decryption key with attributes {PhD, Computer Science} and be able to decrypt ciphertexts in the example above, and the decryption key may not have any identity information. At the same time, as *the decryption privileges corresponding to the common attributes could be shared by multiple users rather than being exclusively owned by one user*, a malicious user with some decryption privileges, shared with multiple users, might have an intention to leak partial or even all the decryption privileges to someone else, for example, for financial gain or for some other incentives, as there is little risk of getting caught. This is referred to as *Malicious Key Delegation*. As a result, it is very crucial for a practical CP-ABE system to defend against malicious key delegation, and to support the traceability of decryption keys. However by far, none of the aforementioned CP-ABE systems is traceable, not even in a white-box model, that is, a malicious user creates a new well-formed decryption key from his own key and tries to prevent anyone from finding out from which decryption key that the new well-formed decryption key is. In other words, *White-Box Traceability* means that given a well-formed decryption key as input a tracing algorithm can trace to the key owner. Due to the strong requirement of well-formed decryption key as input, the white-box traceability is sometimes not considered to be strong enough in practice. A stronger and more practical notion is *Black-Box Traceability*, where given a decryption-device while the decryption algorithm and key may remain unknown, a black-box tracing algorithm, which treats the decryption-device as an oracle, can find out the malicious user whose key must have been used in the construction of the decryption-device. Furthermore, it is desirable if the black-box traceability is *fully collusion-resistant*, that is, the tracing algorithm should work even if an adversary is allowed to access an arbitrary number of keys of its own choice (in other words, when an arbitrary number of malicious users collude) during the construction of the decryption-device. We say that a CP-ABE system supports t-collusion-

[3] The schemes in [21,28] can be extended to allow reuse of attributes by setting a fixed bound on the maximum number of times an attribute may be used and having separate parameters for each use, but this approach will incur a very significant loss in efficiency. The details are referred to [21,22].

resistant black-box traceability, if an adversary is restricted to have no more than t decryption keys when building the decryption-device. Note that collusion-resistant traceability is orthogonal to collusion-resistant security, which is the primary requirement of CP-ABE.

The problem of building a secure CP-ABE scheme resisting malicious key delegation has recently been studied in [17,24,23,25]. The work by Hinek et al. [17] focuses on deterring a user from leaking the decryption keys by embedding the user's "personal information" (such as credit card number) into the decryption keys, and their system relies on a trusted third party that interacts with the user each time when the user wants to decrypt a ciphertext, and which makes their system less practical. The systems in [24,23,25] try to achieve traceability without such a trusted third party. However, as we will review shortly that an *expressive Black-box* Traceable CP-ABE system is yet to be constructed: (1) the ciphertext policies in [24,23] only support a single **AND** gate with wildcard; (2) the traceability in [24] is in the white-box model only; (3) although [23] made attempts to achieve black-box traceability, the traceability is not collusion-resistant, that is, the adversary is not allowed to have more than one decryption key when building the decryption-device, otherwise, their tracing algorithm cannot find out the malicious users; and (4) the traceable CP-ABE scheme in [25] is as expressive (i.e. supporting any monotonic access structures), secure (i.e. provably secure against adaptive adversaries in the standard model) and efficient as the conventional CP-ABE scheme in [21], but the traceability is only in the white-box model.

1.1 Our Results

We propose a *black-box traceable* CP-ABE system whose traceability is fully collusion-resistant and public, that is, anyone can run the tracing algorithm with no additional secret needed or any authority involved. The system is also highly expressive as it supports policies expressed in any monotonic access structures. On its security, we show that it is provably secure and (fully collusion-resistant black-box) traceable against adaptive adversaries in the standard model. Compared with the CP-ABE scheme due to Lewko et al. in [21], i.e. a representative work [4] of the efficient and expressive conventional (non-traceable) CP-ABE systems, our new system *adds* the properties of public and fully collusion-resistant black-box 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 scheme supporting both (public) fully collusion-resistant black-box traceability and high expressiveness simultaneously, and for a system with fully collusion-resistant black-box traceability, sub-linear growth in ciphertext size is the most efficient one to date. Table 1 compares our scheme with that in [21] and [25] in terms of both performance and feature (i.e. traceability), as all the three schemes are secure against adaptive adversaries and highly expressive, that is, supporting any monotonic access structures.

In Sec. 2, we formalize Black-box Traceable CP-ABE (or BT-CP-ABE for short) by *adding* a Trace algorithm to the definition of the conventional CP-ABE. Then, we formalize the full collusion-resistance for black-box traceability using $\mathsf{Game}_{\mathsf{TR}}$: given a decryption-device $\mathcal{D}$, whose decryption

[4] In the most recent work of conventional (non-traceable) CP-ABE due to Lewko and Waters [22], they proposed two CP-ABE schemes, both being expressive, efficient, fully secure in the standard model, and free from the one-use restriction. In particular, they first proposed an construction on composite group which closely resembles the construction of CP-ABE in [21], then obtained a prime order construction by combining their composite order construction and proof with the translation techniques developed by Lewko [20]. Noticed this close resemblance, we expect that our techniques of obtaining traceability are equally applicable to the CP-ABE schemes in [22].

Table 1. Comparison with the conventional CP-ABE in [21] and the traceable CP-ABE in [25]

	Ciphertext Size	Private Key Size	Public Key Size	Pairing Computation in Decryption	Traceability
CP-ABE in [21]	$2L+2$	$\lvert S\rvert+2$	$\lvert\mathbb{S}\rvert+3$	$2\lvert I\rvert+1$	No
CP-ABE in [25]	$2L+3$	$\lvert S\rvert+4$	$\lvert\mathbb{S}\rvert+4$	$2\lvert I\rvert+1$	white-box
Our CP-ABE in this paper	$2L+8\sqrt{\mathcal{K}}$	$\lvert S\rvert+3$	$\lvert\mathbb{S}\rvert+7+8\sqrt{\mathcal{K}}$	$2\lvert I\rvert+5$	public, black-box, fully collusion-resistant

[1] All the three schemes are secure against adaptive adversaries 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, $|\mathbb{S}|$ the size of the attribute universe, and $|I|$ the number of attributes in a key that satisfy the ciphertext's policy.

privilege is described by a non-empty attribute set $S_{\mathcal{D}}$, Trace should be able to extract at least one guilty user whose attribute set is a superset of $S_{\mathcal{D}}$.

On the construction of BT-CP-ABE, instead of building such a system directly, we first build a simpler primitive called Augmented CP-ABE (or AugCP-ABE for short), then we extend it to a BT-CP-ABE system. In Sec. 3.1 we define AugCP-ABE, which is similar to BT-CP-ABE but without the Trace algorithm, namely $(\mathsf{Setup}_\mathsf{A}, \mathsf{KeyGen}_\mathsf{A}, \mathsf{Encrypt}_\mathsf{A}, \mathsf{Decrypt}_\mathsf{A})$, and the encryption algorithm $\mathsf{Encrypt}_\mathsf{A}(\mathsf{PK}, M, \mathbb{A}, \bar{k})$ takes one more parameter $\bar{k} \in \{1, \ldots, \mathcal{K}+1\}$ than $\mathsf{Encrypt}(\mathsf{PK}, M, \mathbb{A})$ does, and the encrypted message M can be recovered using private key $\mathsf{SK}_{k,S}$, which is identified by $k \in \{1, \ldots, \mathcal{K}\}$ and described by an attribute set S, if and only if $(k \geq \bar{k}) \wedge (S \text{ satisfies } \mathbb{A})$. Here $\mathbb{A}$ is an access policy. Also, we define the security model for AugCP-ABE. In Sec. 3.2 we show that a secure AugCP-ABE system can be converted directly to a secure BT-CP-ABE system. In particular, suppose $\Sigma_\mathsf{A} = (\mathsf{Setup}_\mathsf{A}, \mathsf{KeyGen}_\mathsf{A}, \mathsf{Encrypt}_\mathsf{A}, \mathsf{Decrypt}_\mathsf{A})$ is a secure AugCP-ABE system, we set $\mathsf{Encrypt}(\mathsf{PK}, M, \mathbb{A}) = \mathsf{Encrypt}_\mathsf{A}(\mathsf{PK}, M, \mathbb{A}, 1)$, then build the Trace algorithm using a standard technique from the broadcast encryption [7,8,13]. We show that $\Sigma = (\mathsf{Setup}_\mathsf{A}, \mathsf{KeyGen}_\mathsf{A}, \mathsf{Encrypt}, \mathsf{Decrypt}_\mathsf{A}, \mathsf{Trace})$ is a secure BT-CP-ABE system. By taking the approach outlined above, in the rest of the paper, we describe the details of building a secure, expressive and efficient BT-CP-ABE system.

Related Work. In Sec. 6, we go through some related work, which includes various traceability notions in the settings of Key-Policy ABE [34,32], broadcast encryption [11,7,27,26,8,13], and predicate encryption [19].

2 Definitions of Black-box Traceable CP-ABE

A Black-box Traceable CP-ABE (BT-CP-ABE) system is a CP-ABE system with an additional tracing algorithm. Following the notations of the traitor tracing systems in [7,8], a decryption-device (or decoder for short) $\mathcal{D}$ is viewed as a probabilistic circuit that takes as input a ciphertext CT and outputs a message M or $\perp$. In our BT-CP-ABE system, a decoder $\mathcal{D}$ is associated with a non-empty attribute set $S_{\mathcal{D}}$ to describe its decryption ability in that, for the ciphertexts whose policy are satisfied by $S_{\mathcal{D}}$, $\mathcal{D}$ is able to decrypt them correctly with high probability.

Before giving the formal definition of our BT-CP-ABE system, we first explain the reasonability of using a non-empty attribute set to describe a decoder. (1) In practice, if the seller does not explicitly give the decryption ability of a decoder $\mathcal{D}$, the potential buyers are unable to evaluate the price of the decoder and would not buy the decoder. Furthermore, if the decryption ability of a decoder is unknown, the buyer of the decoder has no idea about what ciphertexts he should use the decoder to decrypt. In other words, to some extent, it can be said that a decoder whose decryption ability is unknown is a useless decoder. (2) In different settings of traceability, the decoders are described in different ways. For example, in the Trace and Revoke systems (e.g. [8]) in the setting of broadcast encryption, the decryption ability of a decoder is described by an index set $Y_{\mathcal{D}}$ to imply that the decoder can decrypt a ciphertext if the authorized index set of the ciphertext contains some index in $Y_{\mathcal{D}}$, and in the traceable predicate encryption systems by Katz and Schröder [19], the decryption ability of a decoder is described by a vector $\boldsymbol{v}_{\mathcal{D}}$ to imply that the decoder can decrypt a ciphertext if the ciphertext is encrypted using $\boldsymbol{v}_{\mathcal{D}}$. (3) In the setting of CP-ABE, as the ciphertexts are associated with access policies and the private keys are associated with attribute sets, the natural way of describing the decryption ability of a decoder $\mathcal{D}$ is to give an attribute set $S_{\mathcal{D}}$ implying that $\mathcal{D}$ can decrypt the ciphertexts whose policies can be satisfied by $S_{\mathcal{D}}$. Note that if $S_{\mathcal{D}}$ is empty, $\mathcal{D}$ cannot decrypt any ciphertext and is useless, thus a useful decoder should be described by a non-empty attribute set. In practice, a decoder may be described by multiple non-empty attribute sets, and can decrypt a ciphertext if the associated policy of the ciphertext can be satisfied by any one of the attribute sets. Note that such a decoder can be regarded as being composed of multiple sub-decoders each of which is described by a non-empty attribute set, and our system can be easily extended to handle such a decoder. (4) Another possible way of describing the decryption ability of a decoder $\mathcal{D}$ in the setting of CP-ABE is to give an access policy $\mathbb{A}_{\mathcal{D}}$ implying that $\mathcal{D}$ can decrypt the ciphertexts whose access policies are exactly $\mathbb{A}_{\mathcal{D}}$. Note that this way is unnatural to the setting of CP-ABE and requires the decoders to only have the restricted decryption ability (i.e. assumes that the attackers only construct such restricted decoders). In this paper, we consider the natural way of using a non-empty attribute set to describe the decryption ability of a decoder in CP-ABE, but we believe that our techniques are applicable to the setting where the decoder is described by an access policy and can decrypt only the ciphertexts associated with the same access policy.

A BT-CP-ABE system consists of the following five algorithms:

$\mathsf{Setup}(\lambda, \mathbb{S}, \mathcal{K}) \to (\mathsf{PK}, \mathsf{MSK})$. The algorithm takes as input the security parameter λ, the attribute universe $\mathbb{S}$, and the number of users $\mathcal{K}$ in the system. The algorithm runs in polynomial time in λ and outputs the public parameters PK and a master secret key MSK.

$\mathsf{KeyGen}(\mathsf{PK}, \mathsf{MSK}, S) \to \mathsf{SK}_{k,S}$. The algorithm takes as input the public parameters PK, 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 $k \in \{1, \ldots, \mathcal{K}\}$.

$\mathsf{Encrypt}(\mathsf{PK}, M, \mathbb{A}) \to CT$. The algorithm takes as input the public parameters PK, a message M, and an access policy $\mathbb{A}$ over $\mathbb{S}$, and outputs a ciphertext CT such that only users whose attributes satisfy $\mathbb{A}$ should be able to recover M. $\mathbb{A}$ is implicitly included in CT.

$\mathsf{Decrypt}(\mathsf{PK}, CT, \mathsf{SK}_{k,S}) \to M$ or $\perp$. The algorithm takes as input the public parameters PK, a ciphertext CT associated with an access policy $\mathbb{A}$, and a private key $\mathsf{SK}_{k,S}$. If S satisfies $\mathbb{A}$, the algorithm outputs a message M, otherwise it outputs $\perp$ indicating the failure of decryption.

$\mathsf{Trace}^{\mathcal{D}}(\mathsf{PK}, S_{\mathcal{D}}, \epsilon) \to \mathbb{K}_T \subseteq \{1, \ldots, \mathcal{K}\}$. The algorithm is an oracle algorithm that interacts with a decoder $\mathcal{D}$. The algorithm is given as input the public parameters PK, the associated attribute set $S_{\mathcal{D}}$ of $\mathcal{D}$, and a parameter ϵ, and runs in time polynomial in λ and $1/\epsilon$. Only if $\mathcal{D}$ can decrypt eligible ciphertexts (i.e. the ciphertexts whose policies are satisfied by $S_{\mathcal{D}}$) with probability at least ϵ and ϵ is non-negligible, they are considered valid inputs to Trace. The algorithm outputs an index set $\mathbb{K}_T \subseteq \{1, \ldots, \mathcal{K}\}$ of the guilty users.

We define the security of a BT-CP-ABE system in the following two games. 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 make a private key query for attribute set S the adversary will submit (k, S) to the challenger where k will be the index assigned to the corresponding key.

$\mathsf{Game}_{\mathsf{MH}}$. The first game, called **Message Hiding Game**, is a typical semantic security game, which is almost same to that for the conventional CP-ABE security against adaptive adversaries [21], except that each key is identified by a unique index. In particular, the game is defined between a challenger and an adversary $\mathcal{A}$ (both are given $\mathcal{K}$ and λ as input):

Setup. The challenger runs $\mathsf{Setup}(\lambda, \mathbb{S}, \mathcal{K})$ and gives the public parameters PK to $\mathcal{A}$.
Phase 1. For $i = 1$ to q_1, $\mathcal{A}$ adaptively submits (k_i, S_{k_i}) to the challenger, and the challenger responds with $\mathsf{SK}_{k_i, S_{k_i}}$. Note that $q_1 \leq \mathcal{K}$, $k_i \in \{1, \ldots, \mathcal{K}\}$, and $k_i \neq k_j \ \forall 1 \leq i \neq j \leq q_1$.
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 to $\mathcal{A}$ 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}) to the challenger, and the challenger responds with $\mathsf{SK}_{k_i, S_{k_i}}$. Note that $q \leq \mathcal{K}$, $k_i \in \{1, \ldots, \mathcal{K}\}$, and $k_i \neq k_j \ \forall 1 \leq i \neq j \leq q$.
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 $\mathsf{MHAdv}_{\mathcal{A}} = |\Pr[b' = b] - \frac{1}{2}|$.

Note that the above definitions ensure that a BT-CP-ABE system has all the virtues of a conventional CP-ABE system, and the traceability will be its ***additional*** advantage. We note that the above model can easily be extended to handle chosen-ciphertext attacks (CCA) by allowing for decryption queries in Phase 1 and Phase 2.

$\mathsf{Game}_{\mathsf{TR}}$. The second game captures the notion of **fully collusion-resistant traceability**. The tracing game ensures that the tracing algorithm successfully traces a decoder, no matter how many private keys were used to create the decoder. The adversary's goal is to build a decoder $\mathcal{D}$ that will decrypt the ciphertexts whose policies are satisfied by $S_{\mathcal{D}}$. The tracing algorithm's goal is to extract at least one of the users whose keys have been used for building $\mathcal{D}$. The game is defined between a challenger and an adversary $\mathcal{A}$ (both are given $\mathcal{K}$, λ and ϵ as input) as follows:

Setup. The challenger runs $\mathsf{Setup}(\lambda, \mathbb{S}, \mathcal{K})$ and gives the public parameters PK to $\mathcal{A}$.
Key Query. For $i = 1$ to q, $\mathcal{A}$ adaptively submits (k_i, S_{k_i}) to the challenger, and the challenger responds with $\mathsf{SK}_{k_i, S_{k_i}}$. Note that $q \leq \mathcal{K}$, $k_i \in \{1, \ldots, \mathcal{K}\}$, and $k_i \neq k_j \ \forall 1 \leq i \neq j \leq q$.
Decoder Generation. $\mathcal{A}$ outputs a decoder $\mathcal{D}$ associated with a non-empty attribute set $S_{\mathcal{D}} \subseteq \mathbb{S}$. $\mathcal{D}$ is a probabilistic circuit that takes as input a ciphertext and outputs some message M.
Tracing. The challenger runs $\mathsf{Trace}^{\mathcal{D}}(\mathsf{PK}, S_{\mathcal{D}}, \epsilon)$ 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:

– If an access policy $\mathbb{A}$ is satisfied by $S_{\mathcal{D}}$, then for a randomly chosen message M, we have that

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

A decoder satisfying this condition is said to be a *useful decoder*.

– $\mathbb{K}_T = \emptyset$, 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.

Remark: To be a **traceable** CP-ABE system, when a useful decoder $\mathcal{D}$ is found, the traced $\mathbb{K}_T$ must satisfy $(\mathbb{K}_T \neq \emptyset) \wedge (\mathbb{K}_T \subseteq \mathbb{K}_{\mathcal{D}}) \wedge (\exists k_t \in \mathbb{K}_T\ s.t.\ S_{k_t} \supseteq S_{\mathcal{D}})$. (1) $(\mathbb{K}_T \neq \emptyset) \wedge (\mathbb{K}_T \subseteq \mathbb{K}_{\mathcal{D}})$ captures the preliminary traceability that the tracing algorithm can extract at least one guilty user and the coalition of guilty users cannot frame any innocent user. Note that such a preliminary traceability is a *weak traceability* that is not suitable in practice. Specifically, consider a decoder $\mathcal{D}$ which is built using the private 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 privilege of $\mathcal{D}$ is described by $S_{\mathcal{D}} = S_{k_1}$, e.g., $S_{\mathcal{D}} = S_{k_1} = \{\text{Senior Manager}\}, S_{k_2} = \{\text{Intern}\}$. A scheme is considered to achieve such weak traceability even if its Trace algorithm only extracts k_2 from $\mathcal{D}$ as the guilty user. This is unsatisfying, as $\mathcal{D}$ having the decryption privilege of attribute set {Senior Manager} implies that there must be some user who is authorized attribute "Senior Manager" involved in building $\mathcal{D}$ (this follows from the collusion-resistant security of CP-ABE, which is defined in $\mathsf{Game}_{\mathsf{MH}}$), yet the algorithm was only able to trace $\mathcal{D}$ to an "Intern". Furthermore, in practice a CP-ABE scheme with only weak traceability may suffer from the dilemmas (consider the above example again): the traced user k_2 can argue that in a *secure* CP-ABE system there is no ground to accuse him of building $\mathcal{D}$ because he was authorized only attribute "Intern" and the collusion-resistant security of CP-ABE implies that he does not have the ability to help $\mathcal{D}$ to obtain the decryption privilege of attribute set {Senior Manager}; or the decoder is deliberately designed to sacrifice the low-value user to protect the high-value guilty user when it is necessary as the low-value user often has less to lose even if he is caught. (2) $(\exists k_t \in \mathbb{K}_T\ s.t.\ S_{k_t} \supseteq S_{\mathcal{D}})$ captures the desirable *strong traceability* that the Trace algorithm can extract at least one guilty user whose private key enables $\mathcal{D}$ to have the decryption privilege corresponding to $S_{\mathcal{D}}$, i.e., whose attribute set is a superset of $S_{\mathcal{D}}$. Note that comparable weak and strong traceability notions in the setting of predicate encryption were considered in [19]. While [19] presented formal definitions and constructions for both weak and strong traceability in the setting of predicate encryption, in this paper we focus on only the practical strong traceability of CP-ABE, and unless stated otherwise, by the *traceability* we mean the *strong traceability.*

The tracing game above does not limit the number of colluded users. Furthermore, the decoder does not need to be prefect. It only needs to decrypt with success probability ϵ. Also note that we are modeling a stateless (resettable) decoder – the decoder is just an oracle and maintains no state between activations.

Definition 1. *A $\mathcal{K}$-user Black-box Traceable CP-ABE system is secure if for all polynomial-time adversaries $\mathcal{A}$ the advantages* $\mathsf{MHAdv}_{\mathcal{A}}$ *and* $\mathsf{TRAdv}_{\mathcal{A}}$ *are negligible functions of λ.*

3 Augmented CP-ABE

3.1 Definitions of Augmented CP-ABE

An Augmented CP-ABE (AugCP-ABE) system is a CP-ABE system where each user (private key) is described by an attribute set S and identified by an index k. During encryption, the encryptor encrypts a message with an access policy $\mathbb{A}$ and an index $\bar{k}$. For a user with private key $\mathsf{SK}_{k,S}$, if and only if (S satisfies $\mathbb{A}) \wedge (k \geq \bar{k})$, can he decrypt the corresponding ciphertext. In particular, an AugCP-ABE system consists of the following four algorithms:

$\mathsf{Setup}_{\mathsf{A}}(\lambda, \mathbb{S}, \mathcal{K}) \to (\mathsf{PK}, \mathsf{MSK})$. The algorithm takes as input the security parameter λ, the attribute universe $\mathbb{S}$, and the number of users $\mathcal{K}$ in the system. The algorithm runs in polynomial time in λ and outputs the public parameters PK and a master secret key MSK.

$\mathsf{KeyGen}_{\mathsf{A}}(\mathsf{PK}, \mathsf{MSK}, S) \to \mathsf{SK}_{k,S}$. The algorithm takes as input the public parameters PK, 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 $k \in \{1, \ldots, \mathcal{K}\}$.

$\mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PK}, M, \mathbb{A}, \bar{k}) \to CT$. The algorithm takes as input the public parameters PK, a message M, an access policy $\mathbb{A}$ over $\mathbb{S}$, and an index $\bar{k} \in \{1, \ldots, \mathcal{K}+1\}$, and outputs a ciphertext CT. **$\mathbb{A}$ is included in CT, but the value of $\bar{k}$ is not**.

$\mathsf{Decrypt}_{\mathsf{A}}(\mathsf{PK}, CT, \mathsf{SK}_{k,S}) \to M$ or $\perp$. The algorithm takes as input the public parameters PK, a ciphertext CT associated with an access policy $\mathbb{A}$, and a private key $\mathsf{SK}_{k,S}$. If S satisfies $\mathbb{A}$, the algorithm outputs a message M, otherwise it outputs $\perp$ indicating the failure of decryption.

Correctness. The system must satisfy the following correctness property: for all attribute sets $S \subseteq \mathbb{S}$, all $k \in \{1, \ldots, \mathcal{K}\}$, all access policies $\mathbb{A}$ over $\mathbb{S}$, all $\bar{k} \in \{1, \ldots, \mathcal{K}+1\}$, and all messages M: Let $(\mathsf{PK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}_{\mathsf{A}}(\lambda, \mathbb{S}, \mathcal{K})$, $\mathsf{SK}_{k,S} \leftarrow \mathsf{KeyGen}_{\mathsf{A}}(\mathsf{PK}, \mathsf{MSK}, S)$, $CT \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PK}, M, \mathbb{A}, \bar{k})$. If $(S$ satisfies $\mathbb{A}) \wedge (k \geq \bar{k})$ then $\mathsf{Decrypt}_{\mathsf{A}}(\mathsf{PK}, CT, \mathsf{SK}_{k,S}) = M$.

Security. We define the security of an AugCP-ABE system in the following three games.
The first two games, called **Message Hiding Game**, are defined by the following game for $\bar{k} = 1$ (the first game, $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$) or $\bar{k} = \mathcal{K}+1$ (the second game, $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_{\mathcal{K}+1}}$) between a challenger and an adversary $\mathcal{A}$ (both are given $\mathcal{K}$ and λ as input):

Setup. The challenger runs $\mathsf{Setup}_{\mathsf{A}}(\lambda, \mathbb{S}, \mathcal{K})$ and gives the public parameters PK to $\mathcal{A}$.

Phase 1. For $i = 1$ to q_1, $\mathcal{A}$ adaptively submits (k_i, S_{k_i}) to the challenger, and the challenger responds with $\mathsf{SK}_{k_i, S_{k_i}}$. Note that $q_1 \leq \mathcal{K}$, $k_i \in \{1, \ldots, \mathcal{K}\}$, and $k_i \neq k_j$ $\forall 1 \leq i \neq j \leq q_1$.

Challenge. $\mathcal{A}$ submits two equal-length messages M_0, M_1 and an access policy $\mathbb{A}^*$. The challenger flips a random coin $b \in \{0,1\}$, and sends $CT \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PK}, M_b, \mathbb{A}^*, \bar{k})$ to $\mathcal{A}$.

Phase 2. For $i = q_1 + 1$ to q, $\mathcal{A}$ adaptively submits (k_i, S_{k_i}) to the challenger, and the challenger responds with $\mathsf{SK}_{k_i, S_{k_i}}$. Note that $q \leq \mathcal{K}$, $k_i \in \{1, \ldots, \mathcal{K}\}$, and $k_i \neq k_j$ $\forall 1 \leq i \neq j \leq q$.

Guess. $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ for b.

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

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

$\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$. The third game, called **Index Hiding Game**, says that for any non-empty attribute set $S^* \subseteq \mathbb{S}$, defining **the strictest access policy** $\mathbb{A}_{S^*} = \wedge_{x \in S^*} x$, an adversary cannot distinguish between an encryption using $(\mathbb{A}_{S^*}, \bar{k})$ and one using $(\mathbb{A}_{S^*}, \bar{k}+1)$ without a private 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 $\mathsf{Setup}_{\mathsf{A}}(\lambda, \mathbb{S}, \mathcal{K})$ and gives the public parameters PK to $\mathcal{A}$.

Key Query. For $i = 1$ to q, $\mathcal{A}$ adaptively submits (k_i, S_{k_i}) to the challenger, and the challenger responds with $\mathsf{SK}_{k_i, S_{k_i}}$. Note that $q \leq \mathcal{K}$, $k_i \in \{1, \ldots, \mathcal{K}\}$, and $k_i \neq k_j$ $\forall 1 \leq i \neq j \leq q$.

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 $CT \leftarrow \mathsf{Encrypt_A}(\mathsf{PK}, M, \mathbb{A}_{S^*}, \bar{k}+b)$ 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 $\mathsf{IH^A Adv}_{\mathcal{A}}[\bar{k}] = |\Pr[b' = b] - \frac{1}{2}|$.

Definition 2. *A $\mathcal{K}$-user Augmented CP-ABE system is secure if for all polynomial-time adversaries $\mathcal{A}$ the advantages $\mathsf{MH_1^A Adv}_{\mathcal{A}}$, $\mathsf{MH_{\mathcal{K}+1}^A Adv}_{\mathcal{A}}$ and $\mathsf{IH^A Adv}_{\mathcal{A}}[\bar{k}]$ for $\bar{k} = 1, \ldots, \mathcal{K}$ are negligible functions of λ.*

3.2 Reducing BT-CP-ABE to AugCP-ABE

Let $\Sigma_{\mathsf{A}} = (\mathsf{Setup_A}, \mathsf{KeyGen_A}, \mathsf{Encrypt_A}, \mathsf{Decrypt_A})$ be a secure AugCP-ABE system. Then the derived BT-CP-ABE system is defined as $\Sigma = (\mathsf{Setup_A}, \mathsf{KeyGen_A}, \mathsf{Encrypt}, \mathsf{Decrypt_A}, \mathsf{Trace})$ where

$\mathsf{Encrypt}(\mathsf{PK}, M, \mathbb{A}) = \mathsf{Encrypt_A}(\mathsf{PK}, M, \mathbb{A}, 1)$.

$\mathsf{Trace}^{\mathcal{D}}(\mathsf{PK}, S_{\mathcal{D}}, \epsilon) \to \mathbb{K}_T \subseteq \{1, \ldots, \mathcal{K}\}$: For a given decoder $\mathcal{D}$ associated with a non-empty attribute set $S_{\mathcal{D}}$ and probability $\epsilon > 0$, the tracing algorithm works as follows: [5]

1. For $k = 1$ to $\mathcal{K}+1$, do the following:
   (a) The algorithm repeats the following $8\lambda(\mathcal{K}/\epsilon)^2$ times:
      i. Sample M from the message space at random.
      ii. Let $CT \leftarrow \mathsf{Encrypt_A}(\mathsf{PK}, M, \mathbb{A}_{S_{\mathcal{D}}}, k)$, where $\mathbb{A}_{S_{\mathcal{D}}}$ is the strictest access policy of $S_{\mathcal{D}}$.
      iii. Call oracle $\mathcal{D}$ on input CT which contains $\mathbb{A}_{S_{\mathcal{D}}}$, and compare the output of $\mathcal{D}$ to 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 $\hat{p}_k - \hat{p}_{k+1} \ge \epsilon/(4\mathcal{K})$.
3. Output the set $\mathbb{K}_T$ as the index set of guilty private keys.

Remark: Note that the *strictest access policy* is used in the Index Hiding game $\mathsf{Game_{IH}^A}$ and the tracing algorithm Trace. It is worth noticing that such a strictest access policy is not a limitation of the traceable CP-ABE system. Actually, it is the most efficient way to guarantee that the traced guilty users are the reasonable suspects who have been entitled supersets of $S_{\mathcal{D}}$ and have the ability to construct $\mathcal{D}$. As a decoder $\mathcal{D}$ is designed to decrypt the ciphertexts whose access policy can be satisfied by $S_{\mathcal{D}}$, a ciphertext associated with the strictest access policy $\mathbb{A}_{S_{\mathcal{D}}}$ is a *normal and reasonable* input to $\mathcal{D}$. Although it may be more appealing for the AugCP-ABE scheme to have the index hiding property for any access policy, the following Theorem 1 shows that the strictest access policy is sufficient for guaranteeing the traceability of the derived BT-CP-ABE scheme.

Theorem 1. *If Σ_{A} is a secure AugCP-ABE, then Σ is a secure BT-CP-ABE.*

Proof. Without considering the algorithm Trace, Σ is just a special case of Σ_{A} where the encryption algorithm always sets $\bar{k} = 1$. Consequently, the game $\mathsf{Game_{MH}}$ for Σ is same as the game $\mathsf{Game_{MH_1}^A}$ for Σ_{A}. It implies that the quantity $\mathsf{MHAdv}_{\mathcal{A}}$ for Σ in $\mathsf{Game_{MH}}$ is same with the quantity $\mathsf{MH_1^A Adv}_{\mathcal{A}}$ for Σ_{A} in $\mathsf{Game_{MH_1}^A}$. i.e., if Σ_{A} is a secure AugCP-ABE, then $\mathsf{MHAdv}_{\mathcal{A}}$ is negligible.

Now we show that if Σ_{A} is a secure AugCP-ABE then the quantity $\mathsf{TRAdv}_{\mathcal{A}}$ for Σ in $\mathsf{Game_{TR}}$ is also negligible. In the following proof sketch that is similar to that of [7,8,13], we show that if

[5] The tracing algorithm uses a standard technique from the broadcast encryption [7,8,13].

the decoder output by the adversary is a useful decoder 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 decoder output by the adversary, and $S_{\mathcal{D}}$ be the attribute set describing $\mathcal{D}$. Define

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

We have that $p_1 \geq \epsilon$ and $p_{\mathcal{K}+1}$ is negligible. The former follows from the fact that $\mathcal{D}$ is a useful decoder. The later follows directly from the AugCP-ABE message hiding game $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_{\mathcal{K}+1}}$. 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, the set $\mathbb{K}_T$ output by $\mathsf{Trace}^{\mathcal{D}}(\mathsf{PK}, S_{\mathcal{D}}, \epsilon)$ is non-empty.

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, such a $\mathcal{D}$ can be directly used to win the AugCP-ABE 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 Background

Linear Secret-Sharing Schemes. As of previous work, in this paper we use linear secret-sharing schemes (LSSS) to realize the monotonic access structures that are associated with the ciphertexts to specify the access policies. The formal definitions of access structures and LSSS can be found in Appendix A. Informally, an LSSS is a share-generating matrix A whose rows are labeled by attributes. 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, then $A\boldsymbol{v}$ is the vector of l shares of the secret s. A user's set of attributes S satisfies the LSSS access matrix if the rows labeled by the attributes in S have the *linear reconstruction* property, which means 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. Our AugCP-ABE system works on composite order bilinear groups [6]. Let $\mathcal{G}$ be the group generator, which takes a security parameter λ and outputs $(q, p, p_2, p_3, \mathbb{G}, \mathbb{G}_T, e)$ where q, p, p_2, p_3 are distinct primes, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order $N = qpp_2p_3$, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is 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 $\mathbb{G}_q$, $\mathbb{G}_p$, $\mathbb{G}_{p_2}$ and $\mathbb{G}_{p_3}$ be the subgroups of order q, p, p_2 and p_3 in $\mathbb{G}$, respectively. Note that for any g and h that are from different subgroups of prime order, $e(g, h) = 1$.

Complexity Assumptions. The Message Hiding property of our AugCP-ABE scheme will be based on three assumptions (Assumption 1, 2 and 3 in [21]) that are used by [21] to achieve full security of their CP-ABE system, and the Index Hiding property will be based on three assumptions (Decision (Modified) 3-party Diffie-Hellman Assumption, Diffie-Hellman Subgroup Decision Assumption, and Bilinear Subgroup Decision Assumption in [8]) that are used by [8] to achieve traceability in the setting of broadcast encryption. The details of these assumptions can be found in Appendix B.

5 An Efficient Augmented CP-ABE System

We construct an AugCP-ABE system that is as secure and expressive as the CP-ABE system in [21]. To obtain traceability in the derived BT-CP-ABE system we will use the standard tracing techniques which were used by [7,8,13] in the setting of broadcast encryption. The challenge is to apply the tracing techniques to the setting of CP-ABE *securely and efficiently.*

5.1 Our Approach

In a CP-ABE system, the encryptor will not know or specify the identities of the receivers, so it will be tempting to try naively by combining a Traitor Tracing system (where all the users in the system are always the authorized receivers) and a CP-ABE system for building a CP-ABE system with the traitor tracing property. However, the resulting system cannot achieve the desirable security objective (i.e., *strong* traceability). Consider the following (misguided) approach. Suppose that we created both a CP-ABE system and a Traitor Tracing system each for $\mathcal{K}$ users, where each user 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 the CP-ABE system and M_I under the Traitor Tracing system. In order to decrypt a message a user will need to be able to decrypt under both systems. It is obvious that such an approach can add only weak traceability to the CP-ABE systems [6]. In particular, if two users, Alice with attribute set S_A in the CP-ABE system and index k_A in both systems, and Bob with attribute set S_B in the CP-ABE system and index k_B in both systems, assuming $S_B \cap S_A = \emptyset$, collude to make a decoder $\mathcal{D}$ with attribute set $S_{\mathcal{D}} \subseteq S_A$. The decoder will simply use 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 will be able to only identify Bob as the guilty user involved in building $\mathcal{D}$, although Bob's attribute set S_B is uncorrelated to $S_{\mathcal{D}}$.

The idea behind the techniques of achieving (strong) traceability is to set a user's private key such that it must be *simultaneously* used for both the CP-ABE and the Tracing portions in a traceable CP-ABE. Boneh and Waters handled a similar situation in [8] where they intertwined a Broadcast Encryption system of [5] and a Traitor Tracing system of [7] together to build an Augmented Broadcast Encryption (AugBE) system which implies a Trace and Revoke system. We follow a similar route to their work by applying a tracing technique used in some broadcast encryption systems [7,8,13] to a CP-ABE system of [21]. However, the challenge comes from the setting that in a CP-ABE system the decryption privilege of a user is determined by his attributes rather than by his index as in broadcast encryption systems. In particular, in the AugBE system of [8], each user is identified by an index $k \in \{1, \ldots, \mathcal{K}\}$ which also determines his decryption privilege, i.e., the encryption algorithm $\mathsf{Encrypt}_{\mathsf{AugBE}}(\mathsf{PK}, M, Y, \bar{k})$ will encrypt a message M using an index set $Y \subseteq \{1, \ldots, \mathcal{K}\}$ and an index $\bar{k} \in \{1, \ldots, \mathcal{K}+1\}$ so that a user k can decrypt the ciphertext only if $(k \in Y) \wedge (k \geq \bar{k})$. As Y is an *index set* (for the Broadcast Encryption portion) and $\bar{k}$ is an *index* (for the Tracing portion), they are *essentially correlated*, and the condition $(k \in Y) \wedge (k \geq \bar{k})$ naturally intertwines the two portions together by k. Actually, the construction and proof of Index Hiding property of the AugBE scheme [8] are based on the correlation of $\bar{k}$ and Y in that only if a user possesses a key indexed $\bar{k}$ and $\bar{k} \in Y$ can he distinguish between an encryption to $(Y, \bar{k})$

[6] A similar approach was used in [19] to propose the generic construction of adding weak traceability to any predicate encryption.

and one to $(Y, \bar{k}+1)$. However, the situation in CP-ABE system is fundamentally different. In a traceable CP-ABE system each user is identified by an *index* k that *is uncorrelated to his attribute set* S_k which determines his decryption privilege, i.e., in an AugCP-ABE system the encryption algorithm $\mathsf{Encrypt_A}(\mathsf{PK}, M, \mathbb{A}, \bar{k})$ will encrypt a message M using an access policy $\mathbb{A}$ (defined over attributes) and an index $\bar{k} \in \{1, \ldots, \mathcal{K}+1\}$ so that a user with (k, S_k) can decrypt the ciphertext only if $(S_k$ satisfies $\mathbb{A}) \wedge (k \geq \bar{k})$, where $(S_k,\ \mathbb{A})$ are for the CP-ABE portion and $(k,\ \bar{k})$ for the Tracing portion. As the *index* k and the *attribute set* S_k are *independent from each other*, novelty in construction and proof is needed to correlate $\mathbb{A}$ and $\bar{k}$ and consequently intertwine the CP-ABE and Tracing portions. A straightforward modification and combination will result in schemes that are either not provably secure or inefficient with ciphertext of size $O(\sqrt{\mathcal{K}} \cdot |\mathbb{A}|)$ where $|\mathbb{A}|$ is the size of the access policy. In the following, we propose a *secure* AugCP-ABE system which is also *efficient* with ciphertext of size $O(\sqrt{\mathcal{K}} + |\mathbb{A}|)$.

5.2 Notations

We will express our AugCP-ABE system using the same index notation as the tracing systems [7,8,13] in broadcast encryption. We assume the number of users, $\mathcal{K}$ in the system equals m^2 for some m.[7] We arrange the users in an $m \times m$ matrix. Each user is assigned and identified by a unique tuple (i, j) where $1 \leq i, j \leq m$, and a user in position (i, j) has the index $k = (i-1) * m + j$. For simplicity, we also directly use (i, j) as the index where $(i, j) \geq (\bar{i}, \bar{j})$ means that $(i > \bar{i})$ or $(i = \bar{i} \wedge j \geq \bar{j})$. The use of pairwise notation (i, j) is purely a notational convenience for describing the systems, as $k = (i-1) * m + j$ defines a bijection between $\{(i, j) | 1 \leq i, j \leq m\}$ and $\{1, \ldots, \mathcal{K}\}$.

5.3 AugCP-ABE Construction

$\mathsf{Setup_A}(\lambda, \mathbb{S}, \mathcal{K} = m^2) \rightarrow (\mathsf{PK}, \mathsf{MSK})$. Let $\mathbb{G}$ be a bilinear group of order $N = qpp_2p_3$ (4 distinct primes, whose size is determined by λ), $\mathbb{G}_q, \mathbb{G}_p, \mathbb{G}_{p_2}$ and $\mathbb{G}_{p_3}$ the subgroups of order q, p, p_2 and p_3 respectively in $\mathbb{G}$, and $g_q, h_q, f_q \in \mathbb{G}_q$, $g_p, h_p, f_p \in \mathbb{G}_p$, $X_3 \in \mathbb{G}_{p_3}$ the generators of corresponding subgroups. The algorithm chooses random exponents

$$\beta, \gamma, \delta \in \mathbb{Z}_N, \quad \{r_i,\ z_i,\ \alpha_i \in \mathbb{Z}_N\}_{1 \leq i \leq m}, \quad \{c_j \in \mathbb{Z}_N\}_{1 \leq j \leq m}, \quad \{u_x \in \mathbb{Z}_N\}_{x \in \mathbb{S}}.$$

The public parameters PK include the description of the group and the following elements:

$$\Big(g = g_q g_p,\ h = h_q h_p,\ f = f_q f_p,\ \ G = g^{\delta} g_p^{\gamma},\ H = h^{\delta},\ G_q = g_q^{\beta},\ F_q = f_q^{\beta},$$
$$\{G_i = g^{r_i}, G_{q,i} = g_q^{\beta r_i}, H_i = h^{r_i}, H_{q,i} = h_q^{\beta r_i}, \hat{G}_i = g^{z_i},\ E_i = e(g,g)^{\alpha_i}, E_{q,i} = e(g_q, g_q)^{\beta \alpha_i}\}_{1 \leq i \leq m},$$
$$\{\bar{G}_j = g^{c_j}\}_{1 \leq j \leq m},\ \ \{U_x = g^{u_x}\}_{x \in \mathbb{S}}\ \Big).$$

The master secret key is set to $\mathsf{MSK} = (\ r_1, \ldots, r_m,\ \ c_1, \ldots, c_m,\ \ \alpha_1, \ldots, \alpha_m,\ \ X_3\ )$.
In addition, a counter $ctr = 0$ is implicitly included in MSK.

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

$$\mathsf{SK}_{(i,j),S} = \Big(\ K_{i,j} = g^{\alpha_i} g^{r_i c_j} f^{\sigma_{i,j}} R,\ K'_{i,j} = g^{\sigma_{i,j}} R',\ K''_{i,j} = \hat{G}_i^{\sigma_{i,j}} R'',\ \ \{K_{i,j,x} = U_x^{\sigma_{i,j}} R_x\}_{x \in S}\ \Big).$$

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

$\mathsf{Encrypt_A}(\mathsf{PK}, M, \mathbb{A} = (A, \rho), (\bar{i}, \bar{j})) \to CT$. A is an $L \times D$ LSSS matrix and ρ maps each row A_l of A to an attribute $\rho(l) \in \mathbb{S}$. It is required that ρ would not map two different rows to the same attribute [8]. The algorithm first chooses random exponents

$$\begin{aligned} \kappa,\ \tau,\quad s_{\bar{i}}, \ldots, s_m,\quad d_1, \ldots, d_m,\quad t_1, \ldots, t_m,\quad \mu_1, \ldots, \mu_{\bar{j}-1} &\in \mathbb{Z}_N, \\ (\nu_{1,1}, \nu_{1,2}, \nu_{1,3}), \ldots, (\nu_{\bar{i}-1,1}, \nu_{\bar{i}-1,2}, \nu_{\bar{i}-1,3}) &\in \mathbb{Z}_N^3, \\ (\xi_1, \ldots, \xi_L) \in \mathbb{Z}_N^L, \quad \boldsymbol{v} = (\pi, v_2, \ldots, v_D) &\in \mathbb{Z}_N^D. \end{aligned}$$

Then it creates ciphertext $\langle (A, \rho),\ (R_i, \bar{R}_i, Q_i, \bar{Q}_i, \hat{Q}_i, T_i)_{i=1}^m,\ (C_j, \bar{C}_j)_{j=1}^m,\ (P_l, \bar{P}_l)_{l=1}^L \rangle$ as follows:

1. For each row $i \in \{1, \ldots, m\}$:
   - if $i > \bar{i}$: $R_i = G_{q,i}^{s_i},\ \ \bar{R}_i = H_{q,i}^{\kappa s_i},\ \ Q_i = G_q^{\tau s_i},\ \ \bar{Q}_i = F_q^{\tau s_i} \hat{G}_i^{d_i} f^{\pi},\ \ \hat{Q}_i = g^{d_i},\ \ T_i = M \cdot E_{q,i}^{\tau s_i}$.
   - if $i = \bar{i}$: $R_i = G_i^{s_i},\ \ \bar{R}_i = H_i^{\kappa s_i},\ \ Q_i = g^{\tau s_i},\ \ \bar{Q}_i = f^{\tau s_i} \hat{G}_i^{d_i} f^{\pi},\ \ \hat{Q}_i = g^{d_i},\ \ T_i = M \cdot E_i^{\tau s_i}$.
   - if $i < \bar{i}$: $R_i = g^{\nu_{i,1}},\ \ \bar{R}_i = h^{\kappa \nu_{i,1}},\ \ Q_i = g^{\nu_{i,2}},\ \ \bar{Q}_i = f^{\nu_{i,2}} \hat{G}_i^{d_i} f^{\pi},\ \ \hat{Q}_i = g^{d_i},\ \ T_i = e(g,g)^{\nu_{i,3}}$.
2. For each column $j \in \{1, \ldots, m\}$:
   - if $j \geq \bar{j}$: $C_j = \bar{G}_j^{\tau} h^{\kappa t_j}, \qquad \bar{C}_j = g^{t_j}$.
   - if $j < \bar{j}$: $C_j = \bar{G}_j^{\tau} h^{\kappa t_j} H^{\kappa \mu_j},\ \ \bar{C}_j = g^{t_j} G^{\mu_j}$.
3. For each $l \in \{1, \ldots, L\}$: $\quad P_l = f^{A_l \cdot \boldsymbol{v}} U_{\rho(l)}^{-\xi_l},\ \ \bar{P}_l = g^{\xi_l}$.

$\mathsf{Decrypt_A}(\mathsf{PK}, CT, \mathsf{SK}_{(i,j),S}) \to M$ or $\perp$. The algorithm parses CT to $\langle (A, \rho), (R_i, \bar{R}_i, Q_i, \bar{Q}_i, \hat{Q}_i, T_i)_{i=1}^m$, $(C_j, \bar{C}_j)_{j=1}^m,\ (P_l, \bar{P}_l)_{l=1}^L \rangle$. If S does not satisfy (A, ρ), the algorithm outputs $\perp$, otherwise it

1. Computes constants $\{\omega_l \in \mathbb{Z}_N\}$ such that $\sum_{\rho(l) \in S} \omega_l A_l = (1, 0, \ldots, 0)$, then computes
$$D_P = \prod_{\rho(l) \in S} \big(e(K'_{i,j}, P_l) e(K_{i,j,\rho(l)}, \bar{P}_l) \big)^{\omega_l} = \prod_{\rho(l) \in S} \big(e(g^{\sigma_{i,j}}, f^{A_l \cdot \boldsymbol{v}}) \big)^{\omega_l} = e(g^{\sigma_{i,j}}, f)^{\pi}.$$
2. Computes $D_I = \big(e(K_{i,j}, Q_i) \cdot e(K''_{i,j}, \hat{Q}_i) \cdot e(\bar{R}_i, \bar{C}_j) \big) / \big(e(K'_{i,j}, \bar{Q}_i) \cdot e(R_i, C_j) \big)$.
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 C.

Remarks: (1) In the Tracing portion of the ciphertext, while $(R_i, \bar{R}_i, Q_i, T_i,\ C_j, \bar{C}_j)$ are same to that of [8], $\bar{Q}_i$ is different and $\hat{Q}_i$ is a new component we design. $\hat{G}_i^{d_i}$ (in $\bar{Q}_i$) is the crucial component that *intertwines* the Tracing portion ($F_q^{\tau s_i}$ for $i > \bar{i}$ and $f^{\tau s_i}$ for $i = \bar{i}$) and the CP-ABE portion (f^{π}) *securely and efficiently.* In a straightforward combination without $\hat{G}_i^{d_i}$ (i.e., $\bar{Q}_i = F_q^{\tau s_i} f^{\pi}$ for $i > \bar{i}$, and $\bar{Q}_i = f^{\tau s_i} f^{\pi}$ for $i = \bar{i}$), the index hiding property will be hard to prove. And to obtain provable index hiding property, different π_i has to be used for different i (i.e., $\bar{Q}_i = F_q^{\tau s_i} f^{\pi_i}$ for $i > \bar{i}$, and $\bar{Q}_i = f^{\tau s_i} f^{\pi_i}$ for $i = \bar{i}$), so that the CP-ABE portion of the resulting system will have ciphertext size of $O(\sqrt{\mathcal{K}} \cdot L)$, rather than $O(L)$ as above. (2) In the construction above, the size of the public parameters grows linear with $|\mathbb{S}|$, the size of the attribute universe. Note that using the techniques in [21] we can obtain a large attribute universe construction as [21] does, where the size of public parameters is independent of $|\mathbb{S}|$.

[8] This restriction is inherited from the underlying CP-ABE scheme in [21] whose security proof relies on such a restriction, and can be removed with the techniques in [21] similarly. In the most recent (non-traceable) CP-ABE schemes in [22], this restriction is eliminated using a new proof technique, but the (composite order) construction in [22] still closely resembles that of [21], so we expect that our techniques of obtaining traceability are applicable to the new CP-ABE schemes in [22].

5.4 AugCP-ABE Security

The security of our AugCP-ABE construction follows from the following Theorem 2, 3 and 4.

Theorem 2. *Suppose that Assumptions 1, 2, and 3 in [21] hold. Then no polynomial time adversary can win* $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$ *with non-negligible advantage.*

Proof. The CP-ABE scheme in [21] is proved secure against adaptive adversaries in the standard model, based on Assumptions 1, 2, and 3. Note that the structures of CP-ABE portion of our AugCP-ABE scheme are similar to that of the CP-ABE scheme in [21], this theorem can be proved using similar proof. The proof details can be found in Appendix D.1.

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

Proof. The argument for security of $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_{\mathcal{K}+1}}$ 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 $\mathcal{K}+1$. Since for all $i = 1$ to m, the values of $T_i = e(g,g)^{\nu_{i,3}}$ 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 Decision (Modified) 3-party Diffie-Hellman, Bilinear Subgroup Decision, and Diffie-Hellman Subgroup Decision assumptions 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 Decision (Modified) 3-party Diffie-Hellman Assumption holds. Then for* $\bar{j} < m$ *no polynomial time adversary can distinguish between an encryption to* $(\bar{i}, \bar{j})$ *and an encryption to* $(\bar{i}, \bar{j}+1)$ *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 obtain the corresponding private key. In Challenge phase, $\mathcal{A}$ sends a message M and an attribute set S^* to $\mathcal{B}$. 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 obtain the corresponding private key. In Challenge phase, $\mathcal{A}$ sends a message M and an attribute set S^* to $\mathcal{B}$ with the restriction that $S_{(\bar{i},\bar{j})}$ does not satisfy $\mathbb{A}_{S^*}$ (i.e., $S^* \setminus S_{(\bar{i},\bar{j})} \neq \emptyset$).

At the beginning of the game, the simulator $\mathcal{B}$ does not know which case $\mathcal{A}$ will behave in. $\mathcal{B}$ will guess it by flipping a random coin $\tilde{b} \in \{0,1\}$. We will show that in both cases $\mathcal{B}$'s output will be the same as that in the real game, i.e., the value of $\tilde{b}$ is hidden from $\mathcal{A}$. Then $\mathcal{B}$ can finish the simulation with probability 1/2.

The simulation for **Case I** is very similar to that of [8] because the simulator does not need to generate private key indexed $(\bar{i}, \bar{j})$ and there is not any restriction on the attribute set S^*.

The **Case II** captures the security that even when a user has a key indexed $(\bar{i}, \bar{j})$ he cannot distinguish between an encryption to $(\mathbb{A}_{S^*}, (\bar{i}, \bar{j}))$ and one to $(\mathbb{A}_{S^*}, (\bar{i}, \bar{j}+1))$ if the corresponding attribute set $S_{(\bar{i},\bar{j})}$ is not a superset of S^*. With the crucial components $\hat{G}_i^{d_i}$ (in $\bar{Q}_i$) and $\hat{Q}_i = g^{d_i}$ in the ciphertext, our particular construction guarantees that $\mathcal{B}$ can successfully finish the simulation with probability $|S^* \setminus S_{(\bar{i},\bar{j})}|/|\mathbb{S}|$, which is at least $1/|\mathbb{S}|$ since $S^* \setminus S_{(\bar{i},\bar{j})} \neq \emptyset$.

The proof details of Lemma 1 can be found in Appendix D.2.

Lemma 2. *Suppose that the Decision (Modified) 3-party Diffie-Hellman, Bilinear Subgroup Decision, and Diffie-Hellman Subgroup Decision assumptions hold. Then no polynomial time adversary can distinguish between an encryption to* $(\bar{i}, m)$ *and one to* $(\bar{i}+1, 1)$ *in* $\mathsf{Game}_{\mathsf{IH}}^{\mathsf{A}}$ *with non-negligible advantage.*

Proof. Similar to the proof of Lemma 5.3 in [8], 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 Decision (Modified) 3-party Diffie-Hellman assumption holds. Then no polynomial time adversary can distinguish between experiments* H_1 *and* H_2 *with non-negligible advantage.*

Proof. The proof of Claim 1 is identical to that of Lemma 1.

Claim 2. *Suppose that the Decision (Modified) 3-party Diffie-Hellman, Bilinear Subgroup Decision, and Diffie-Hellman Subgroup Decision assumptions 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 similar proof to that of Claim 5.5, 5.6 and 5.7 in [8], which were used to prove the indistinguishability of similar hybrid experiments for their Augmented Broadcast Encryption (AugBE) scheme. The proof details can be found in Appendix D.3.

Remarks: (1) Our construction works on a composite order group, and the security is based on six assumptions. Recently, Lewko [20] proposed a general methodology for converting composite order pairing-based cryptosystems into the prime order setting, and the security of the resulting systems is based on a standard assumption over prime order group (Decisional Linear Assumption). We note that Lewko's technique is applicable to our construction. The resulting construction will work on a prime order group, and the security will be based on two standard assumptions (Decisional Linear Assumption and Decision 3-party Diffie Hellman Assumption). (2) While we proved our construction secure under chosen-plaintext attacks, it is not difficult to modify it slightly and apply the methods of Canetti, Halevi, and Katz [9] for security against chosen-ciphertext attacks.

6 Related Work

Sahai and Waters [30] introduced Attribute-Based Encryption (ABE) for addressing the fuzzy identity matching problem in IBE. Goyal et al. [15] later formalized the notions of CP-ABE and Key-Policy ABE (KP-ABE). In a KP-ABE system, each ciphertext comes with a set of attributes and each user has a decryption key associated with an access policy issued by an authority. Applications of KP-ABE include "audit-log" and "pay-TV" [15], and KP-ABE systems available in the literature include [29,21,28,1], however, these system do not address the malicious key delegation problem. [34,32] are among the first KP-ABE systems to consider traceability. The system in [34] supports expressive key-policy, but its black-box traceability is not collusion-resistant. Supporting only limited key-policy (i.e., a single threshold gate), the system in [32] achieves t-collusion-resistant black-box traceability at the expense of adding overhead in the order of $O(t^2 \log \mathcal{K})$ to the underlying system [30], where $\mathcal{K}$ is the number of users in the system. Note that to achieve *fully collusion-resistant* traceability, t is equal to $\mathcal{K}$, and therefore, the overhead will be in the order of $O(\mathcal{K}^2 \log \mathcal{K})$, which is very big and might not be practical for large $\mathcal{K}$.

In broadcast encryption, traceability is also an important feature. There are three types of systems: (1) in a Broadcast Encryption [12,5] system, a broadcaster encrypts messages for *an arbitrary subset* $Y \subseteq \{1, \ldots, \mathcal{K}\}$ of users in the system; (2) in a Traitor Tracing [11,7] system, a broadcaster encrypts messages so that *all* $\mathcal{K}$ users can decrypt, while if some of the users collude and jointly build a pirate decryption-device $\mathcal{D}$, a tracing algorithm Trace by taking $\mathcal{D}$ as an input can output the identity of *at least one* of the malicious users; (3) a Trace and Revoke [27,26,8,13] system provides both Broadcast Encryption and Traitor Tracing. Note that achieving fully collusion-resistant black-box traceability at the expense of sub-linear size ciphertext is the best result currently available for broadcast encryption, and our attempt here is to achieve a comparable result in the setting of CP-ABE, which is very different from that of broadcast encryption.

Katz and Schröder [19] introduced the notion of traceability in the context of predicate encryption [18]. In a predicate encryption scheme, each ciphertext is associated (by the encryptor) with an attribute and each user has a decryption key associated with a predicate issued by an authority, and a ciphertext associated with attribute I can be decrypted using a decryption key associated with predicate f only if $f(I) = 1$. [19] proposed a generic construction that adds traceability to any inner-product predicate encryption (IPE) scheme. Note that although IPE (e.g., the most expressive schemes to date in [18]) is general enough to cover IBE, BE and KP-ABE, it does not cover CP-ABE, especially expressive CP-ABE. Also note that the construction of [19] adds overhead linear in $\mathcal{K}$ (the number of users) to the original scheme and it is left as an open problem to propose more efficient constructions even for specific IPE schemes. The advances of our paper 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 [19] is.

7 Conclusion

We constructed the first expressive Black-box Traceable CP-ABE system that simultaneously supports fully collusion-resistant (and public) black-box traceability and high expressiveness (i.e., supporting any monotonic access structures). The system is proven secure and traceable against adaptive adversaries in the standard model. Compared with the representative work of the efficient conventional (non-traceable) CP-ABE systems currently available for high expressiveness, our new CP-ABE system adds fully collusion-resistant black-box traceability with the price of adding only $O(\sqrt{\mathcal{K}})$ elements in the ciphertext and public key. Instead of directly building a Black-box Traceable CP-ABE system, we constructed a simpler primitive called Augmented CP-ABE, and showed that a secure Augmented CP-ABE system is sufficient for constructing a Black-box Traceable CP-ABE system with fully collusion-resistant traceability.

References

1. Attrapadung, N., Libert, B., de Panafieu, E.: Expressive key-policy attribute-based encryption with constant-size ciphertexts. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) Public Key Cryptography. LNCS, vol. 6571, pp. 90–108. Springer (2011)
2. Beimel, A.: Secure Schemes for Secret Sharing and Key Distribution. Ph.D. thesis, Israel Institute of Technology, Technion, Haifa, Israel (1996)

3. 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)
4. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO. LNCS, vol. 2139, pp. 213–229. Springer (2001)
5. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: Shoup, V. (ed.) CRYPTO. LNCS, vol. 3621, pp. 258–275. Springer (2005)
6. Boneh, D., Goh, E.J., Nissim, K.: Evaluating 2-dnf formulas on ciphertexts. In: Kilian, J. (ed.) TCC. LNCS, vol. 3378, pp. 325–341. Springer (2005)
7. Boneh, D., Sahai, A., Waters, B.: Fully collusion resistant traitor tracing with short ciphertexts and private keys. In: Vaudenay, S. (ed.) EUROCRYPT. LNCS, vol. 4004, pp. 573–592. Springer (2006)
8. Boneh, D., Waters, B.: A fully collusion resistant broadcast, trace, and revoke system. In: Juels, A., Wright, R.N., di Vimercati, S.D.C. (eds.) ACM CCS. pp. 211–220. ACM (2006)
9. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT. LNCS, vol. 3027, pp. 207–222. Springer (2004)
10. Cheung, L., Newport, C.C.: Provably secure ciphertext policy abe. In: Ning, P., di Vimercati, S.D.C., Syverson, P.F. (eds.) ACM CCS. pp. 456–465. ACM (2007)
11. Chor, B., Fiat, A., Naor, M.: Tracing traitors. In: Desmedt, Y. (ed.) CRYPTO. LNCS, vol. 839, pp. 257–270. Springer (1994)
12. Fiat, A., Naor, M.: Broadcast encryption. In: Stinson, D.R. (ed.) CRYPTO. LNCS, vol. 773, pp. 480–491. Springer (1993)
13. Garg, S., Kumarasubramanian, A., Sahai, A., Waters, B.: Building efficient fully collusion-resilient traitor tracing and revocation schemes. In: Al-Shaer, E., Keromytis, A.D., Shmatikov, V. (eds.) ACM CCS. pp. 121–130. ACM (2010)
14. Goyal, V., Jain, A., Pandey, O., Sahai, A.: Bounded ciphertext policy attribute based encryption. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP (2). LNCS, vol. 5126, pp. 579–591. Springer (2008)
15. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Juels, A., Wright, R.N., di Vimercati, S.D.C. (eds.) ACM CCS. pp. 89–98. ACM (2006)
16. Herranz, J., Laguillaumie, F., Ràfols, C.: Constant size ciphertexts in threshold attribute-based encryption. In: Nguyen, P.Q., Pointcheval, D. (eds.) Public Key Cryptography. LNCS, vol. 6056, pp. 19–34. Springer (2010)
17. Hinek, M.J., Jiang, S., Safavi-Naini, R., Shahandashti, S.F.: Attribute-based encryption with key cloning protection. Cryptology ePrint Archive, Report 2008/478 (2008), http://eprint.iacr.org/
18. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart, N.P. (ed.) EUROCRYPT. LNCS, vol. 4965, pp. 146–162. Springer (2008)
19. Katz, J., Schröder, D.: Tracing insider attacks in the context of predicate encryption schemes. In: ACITA (2011), https://www.usukita.org/node/1779
20. Lewko, A.B.: Tools for simulating features of composite order bilinear groups in the prime order setting. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT. LNCS, vol. 7237, pp. 318–335. Springer (2012)
21. Lewko, A.B., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In: Gilbert, H. (ed.) EUROCRYPT. LNCS, vol. 6110, pp. 62–91. Springer (2010)
22. Lewko, A.B., Waters, B.: New proof methods for attribute-based encryption: Achieving full security through selective techniques. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO. Lecture Notes in Computer Science, vol. 7417, pp. 180–198. Springer (2012)
23. 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: Cheung, B.S.N., Hui, L.C.K., Sandhu, R.S., Wong, D.S. (eds.) ASIACCS. pp. 386–390. ACM (2011)
24. Li, J., Ren, K., Kim, K.: A2be: Accountable attribute-based encryption for abuse free access control. Cryptology ePrint Archive, Report 2009/118 (2009), http://eprint.iacr.org/
25. 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) (2013), available at http://dx.doi.org/10.1109/TIFS.2012.2223683
26. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: Kilian, J. (ed.) CRYPTO. LNCS, vol. 2139, pp. 41–62. Springer (2001)
27. Naor, M., Pinkas, B.: Efficient trace and revoke schemes. In: Frankel, Y. (ed.) Financial Cryptography. LNCS, vol. 1962, pp. 1–20. Springer (2000)

28. Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: Rabin, T. (ed.) CRYPTO. LNCS, vol. 6223, pp. 191–208. Springer (2010)
29. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: Ning, P., di Vimercati, S.D.C., Syverson, P.F. (eds.) ACM CCS. pp. 195–203. ACM (2007)
30. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT. LNCS, vol. 3494, pp. 457–473. Springer (2005)
31. Shamir, A.: Identity-based cryptosystems and signature schemes. In: CRYPTO. pp. 47–53 (1984)
32. Wang, Y.T., Chen, K.F., Chen, J.H.: Attribute-based traitor tracing. J. Inf. Sci. Eng. 27(1), 181–195 (2011)
33. Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) Public Key Cryptography. LNCS, vol. 6571, pp. 53–70. Springer (2011)
34. Yu, S., Ren, K., Lou, W., Li, J.: Defending against key abuse attacks in kp-abe enabled broadcast systems. In: Chen, Y., Dimitriou, T., Zhou, J. (eds.) SecureComm. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 19, pp. 311–329. Springer (2009)

A Access Policy

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

In ABE systems, the role of the parties is taken by the attributes. Thus, the access structure $\mathbb{A}$ contains the authorized sets of attributes. As of previous work, we focus on monotone access structures in this paper. It is shown in [2] that any monotone access structure can be realized by a linear secret sharing scheme. Here we use the definition from [2,33].

Definition 4 (Linear Secret-Sharing Schemes (LSSS) [33]). *A secret sharing scheme Π over a set of parties $\mathbb{P}$ is called linear (over $\mathbb{Z}_p$) if*

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

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

B Assumptions

In Assumptions 1, 2, and 3 below, $\mathbb{G}$ is a bilinear group of order $N = qp_2p_3$ where q, p_2, p_3 are distinct primes. Let $\mathbb{G}_q$, $\mathbb{G}_{p_2}$, $\mathbb{G}_{p_3}$, $\mathbb{G}_{qp_2}$ and $\mathbb{G}_{qp_3}$ be the subgroups of order q, p_2, p_3, qp_2 and qp_3 in $\mathbb{G}$, respectively.

Assumption 1 (Subgroup decision problem for 3 primes) *[21] Given a group generator $\mathcal{G}$, define the following distribution: $(q, p_2, p_3, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}$, $N = qp_2p_3$, $g \xleftarrow{R} \mathbb{G}_q$, $X_3 \xleftarrow{R} \mathbb{G}_{p_3}$, $D = ((N, \mathbb{G}, \mathbb{G}_T, e), g, X_3)$, $T_1 \xleftarrow{R} \mathbb{G}_{qp_2}$, $T_2 \xleftarrow{R} G_q$. The advantage of an algorithm $\mathcal{A}$ in breaking Assumption 1 is:*

$$Adv1_{\mathcal{G},\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|.$$

Definition 5. *$\mathcal{G}$ satisfies Assumption 1 if $Adv1_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ for any polynomial time algorithm $\mathcal{A}$.*

Assumption 2 *[21] Given $\mathcal{G}$, define the following distribution: $(q, p_2, p_3, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}$, $N = qp_2p_3$, $g, X_1 \xleftarrow{R} \mathbb{G}_q$, $X_2, Y_2 \xleftarrow{R} \mathbb{G}_{p_2}$, $X_3, Y_3 \xleftarrow{R} \mathbb{G}_{p_3}$, $D = ((N, \mathbb{G}, \mathbb{G}_T, e), g, X_1X_2, X_3, Y_2Y_3), T_1 \xleftarrow{R} \mathbb{G}$, $T_2 \xleftarrow{R} \mathbb{G}_{qp_3}$. The advantage of an algorithm $\mathcal{A}$ in breaking Assumption 2 is:*

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

Definition 6. *$\mathcal{G}$ satisfies Assumption 2 if $Adv2_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ for any polynomial time algorithm $\mathcal{A}$.*

Assumption 3 *[21] Given $\mathcal{G}$, define the following distribution: $(q, p_2, p_3, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}$, $N = qp_2p_3$, $\alpha, s \xleftarrow{R} \mathbb{Z}_N$, $g \xleftarrow{R} \mathbb{G}_q$, $X_2, Y_2, Z_2 \xleftarrow{R} \mathbb{G}_{p_2}$, $X_3 \xleftarrow{R} \mathbb{G}_{p_3}$, $D = ((N, \mathbb{G}, \mathbb{G}_T, e), g, g^{\alpha}X_2, X_3, g^sY_2, Z_2)$, $T_1 = e(g, g)^{\alpha s}$, $T_2 \xleftarrow{R} \mathbb{G}_T$. The advantage of an algorithm $\mathcal{A}$ in breaking Assumption 3 is:*

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

Definition 7. *$\mathcal{G}$ satisfies Assumption 3 if $Adv3_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ for any polynomial time algorithm $\mathcal{A}$.*

Assumption 4 (Decision (Modified) 3-party Diffie-Hellman Assumption) *[8] Given a group generator $\mathcal{G}$, define the following distribution: $(p, \mathbb{G}_p, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}$, $g_p \xleftarrow{R} \mathbb{G}_p$, $a, b, c \xleftarrow{R} \mathbb{Z}_p$, $D = ((p, \mathbb{G}_p, \mathbb{G}_T, e),\ g_p, g_p^a, g_p^b, g_p^c, g_p^{b^2})$, $T = g_p^{abc}$, $R \xleftarrow{R} \mathbb{G}_p$. The advantage of an algorithm $\mathcal{A}$ in breaking the decision 3-party Diffie-Hellman Assumption is:*

$$Adv\mathsf{D3DH}_{\mathcal{G},\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(D, T) = 1] - \Pr[\mathcal{A}(D, R) = 1]|.$$

Definition 8. *$\mathcal{G}$ satisfies the Decision (Modified) 3-party Diffie-Hellman Assumption if $Adv\mathsf{D3DH}_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ for any polynomial time algorithm $\mathcal{A}$.*

Assumption 5 (Diffie-Hellman Subgroup Decision Assumption) *[8] Given a group generator $\mathcal{G}$, define the following distribution: $(q, p, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}$, $N = qp$, $g, h \xleftarrow{R} \mathbb{G}$, $v_p \xleftarrow{R} \mathbb{G}_p$, $a \xleftarrow{R} \mathbb{Z}_q, b \xleftarrow{R} \mathbb{Z}_N$, $D = ((N, \mathbb{G}, \mathbb{G}_T, e), g, h, g^b v_p, h^b, g^{pa}, h^{pa})$, $T_1 \xleftarrow{R} \mathbb{G}_q$, $T_2 \xleftarrow{R} \mathbb{G}$. The advantage of an algorithm $\mathcal{A}$ in breaking the Diffie-Hellman Subgroup Decision Assumption is:*

$$Adv\mathsf{DHSD}_{\mathcal{G},\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|.$$

Definition 9. *$\mathcal{G}$ satisfies the Diffie-Hellman Subgroup Decision Assumption if $Adv\mathsf{DHSD}_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ for any polynomial time algorithm $\mathcal{A}$.*

Assumption 6 (Bilinear Subgroup Decision Assumption) *[8] Given a group generator $\mathcal{G}$, define the following distribution: $(q, p, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}$, $N = qp$, $g_q \xleftarrow{R} \mathbb{G}_q$, $g_p \xleftarrow{R} \mathbb{G}_p$, $g = g_q g_p$, $T \xleftarrow{R} \mathbb{G}_p$, $R \xleftarrow{R} \mathbb{G}$, $D = ((N, \mathbb{G}, \mathbb{G}_T, e), g_p, g_q)$, $T_1 = e(g, T)$, $T_2 = e(g, R)$. The advantage of an algorithm $\mathcal{A}$ in breaking the Bilinear Subgroup Decision Assumption is:*

$$Adv\mathsf{BSD}_{\mathcal{G},\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|.$$

Definition 10. *$\mathcal{G}$ satisfies the Bilinear Subgroup Decision Assumption if $Adv\mathsf{BSD}_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ for any polynomial time algorithm $\mathcal{A}$.*

C Correctness of Our AugCP-ABE Construction

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

- If $i > \bar{i}, j \geq \bar{j}$,

$$\begin{aligned} D_I =& \big(e(g^{\alpha_i} g^{r_i c_j} f^{\sigma_{i,j}}, G_q^{\tau s_i}) e(\hat{G}_i^{\sigma_{i,j}}, g^{d_i}) e(H_{q,i}^{\kappa s_i}, g^{t_j})\big) / \big(e(g^{\sigma_{i,j}}, F_q^{\tau s_i} \hat{G}_i^{d_i} f^{\pi}) e(G_{q,i}^{s_i}, \bar{G}_j^{\tau} h^{\kappa t_j})\big) \\ =& \big(e(g^{\alpha_i} g^{r_i c_j} f^{\sigma_{i,j}}, g_q^{\beta \tau s_i}) e(h_q^{\beta r_i \kappa s_i}, g^{t_j})\big) / \big(e(g^{\sigma_{i,j}}, f_q^{\beta \tau s_i} f^{\pi}) e(g_q^{\beta r_i s_i}, g^{c_j \tau} h^{\kappa t_j})\big) \\ =& e(g, g_q)^{\beta \alpha_i \tau s_i} / e(g^{\sigma_{i,j}}, f^{\pi}) \end{aligned}$$

- If $i > \bar{i}, j < \bar{j}$,

$$\begin{aligned} D_I =& \big(e(g^{\alpha_i} g^{r_i c_j} f^{\sigma_{i,j}}, G_q^{\tau s_i}) e(\hat{G}_i^{\sigma_{i,j}}, g^{d_i}) e(H_{q,i}^{\kappa s_i}, g^{t_j} G^{\mu_j})\big) / \big(e(g^{\sigma_{i,j}}, F_q^{\tau s_i} \hat{G}_i^{d_i} f^{\pi}) e(G_{q,i}^{s_i}, \bar{G}_j^{\tau} h^{\kappa t_j} H^{\kappa \mu_j})\big) \\ =& \big(e(g^{\alpha_i} g^{r_i c_j} f^{\sigma_{i,j}}, g_q^{\beta \tau s_i}) e(h_q^{\beta r_i \kappa s_i}, g^{t_j} g^{\delta \mu_j} g_p^{\gamma \mu_j})\big) / \big(e(g^{\sigma_{i,j}}, f_q^{\beta \tau s_i} f^{\pi}) e(g_q^{\beta r_i s_i}, g^{c_j \tau} h^{\kappa t_j} h^{\delta \kappa \mu_j})\big) \\ =& e(g, g_q)^{\beta \alpha_i \tau s_i} / e(g^{\sigma_{i,j}}, f^{\pi}) \end{aligned}$$

- If $i = \bar{i}, j \geq \bar{j}$,

$$\begin{aligned} D_I =& \big(e(g^{\alpha_i} g^{r_i c_j} f^{\sigma_{i,j}}, g^{\tau s_i}) e(\hat{G}_i^{\sigma_{i,j}}, g^{d_i}) e(H_i^{\kappa s_i}, g^{t_j})\big) / \big(e(g^{\sigma_{i,j}}, f^{\tau s_i} \hat{G}_i^{d_i} f^{\pi}) e(G_i^{s_i}, \bar{G}_j^{\tau} h^{\kappa t_j})\big) \\ =& \big(e(g^{\alpha_i} g^{r_i c_j} f^{\sigma_{i,j}}, g^{\tau s_i}) e(h^{r_i \kappa s_i}, g^{t_j})\big) / \big(e(g^{\sigma_{i,j}}, f^{\tau s_i} f^{\pi}) e(g^{r_i s_i}, g^{c_j \tau} h^{\kappa t_j})\big) \\ =& e(g, g)^{\alpha_i \tau s_i} / e(g^{\sigma_{i,j}}, f^{\pi}) \end{aligned}$$

- If $i = \bar{i}, j < \bar{j}$,

$$\begin{aligned} D_I =& \big(e(g^{\alpha_i} g^{r_i c_j} f^{\sigma_{i,j}}, g^{\tau s_i}) e(\hat{G}_i^{\sigma_{i,j}}, g^{d_i}) e(H_i^{\kappa s_i}, g^{t_j} G^{\mu_j})\big) / \big(e(g^{\sigma_{i,j}}, f^{\tau s_i} \hat{G}_i^{d_i} f^{\pi}) e(G_i^{s_i}, \bar{G}_j^{\tau} h^{\kappa t_j} H^{\kappa \mu_j})\big) \\ =& \big(e(g^{\alpha_i} g^{r_i c_j} f^{\sigma_{i,j}}, g^{\tau s_i}) e(h^{r_i \kappa s_i}, g^{t_j} g^{\delta \mu_j} g_p^{\gamma \mu_j})\big) / \big(e(g^{\sigma_{i,j}}, f^{\tau s_i} f^{\pi}) e(g^{r_i s_i}, g^{c_j \tau} h^{\kappa t_j} h^{\delta \kappa \mu_j})\big) \\ =& e(g, g)^{\alpha_i \tau s_i} \cdot e(h, g_p)^{r_i \kappa s_i \gamma \mu_j} / e(g^{\sigma_{i,j}}, f^{\pi}) \end{aligned}$$

Then from the values of T_i, D_P and D_I, we have that

$$M' = T_i / (D_P \cdot D_I) = \begin{cases} M, & \text{if } i > \bar{i} \wedge j \geq \bar{j}, \\ M, & \text{if } i > \bar{i} \wedge j < \bar{j}, \\ M, & \text{if } i = \bar{i} \wedge j \geq \bar{j}, \\ M \cdot e(h, g_p)^{-r_i \kappa s_i \gamma \mu_j}, & \text{if } i = \bar{i} \wedge j < \bar{j}, \\ \text{has no relation with } M, & \text{if } i < \bar{i}. \end{cases}$$

D Proofs

D.1 Proof of Theorem 2

Note that the structures of CP-ABE portion of our AugCP-ABE scheme are similar to that of the CP-ABE scheme in [21], the proof of Theorem 2 is also similar to that of [21]. Due to the space restriction and for simplicity, here we prove the theorem by reducing message hiding security of our AugCP-ABE scheme in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$ to the security of CP-ABE scheme in [21]. Theorem 2 follows from the following Lemma 3 and Lemma 4.

Lemma 3. *[21] If Assumptions 1, 2, and 3 hold, then the CP-ABE scheme in [21] is secure.*

Proof. This lemma follows from the Theorem 1 of [21] immediately.

Lemma 4. *Suppose the CP-ABE scheme in [21] is secure. Then for our AugCP-ABE scheme no polynomial time adversary can win the Message Hiding Game* $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$ *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 $\mathsf{MH}^{\mathsf{A}}_1\mathsf{Adv}_{\mathcal{A}}$, we construct a PPT algorithm $\mathcal{B}$ to break the CP-ABE scheme (denoted by Σ_{cpabe}) in [21] with advantage $Adv_{\mathcal{B}}\Sigma_{\mathsf{cpabe}}$, which equals to $\mathsf{MH}^{\mathsf{A}}_1\mathsf{Adv}_{\mathcal{A}}$.

The game of $\mathcal{B}$ attacking Σ_{cpabe} is played in the subgroup $\mathbb{G}_{\tilde{N}}$ of order $\tilde{N} = qp_2p_3$ in a composite group $\mathbb{G}_N$ of order $N = qpp_2p_3$. $\mathcal{B}$ is given the values of $\tilde{N}$ and p, [9] but does not know the values of q, p_2 or p_3. Since the game is played in the subgroup $\mathbb{G}_{\tilde{N}}$, $\mathcal{B}$ chooses for itself everything in the subgroup $\mathbb{G}_p$ in the following simulation.

Setup. $\mathcal{B}$ receives the public parameters [10] $\mathsf{PK}^{\mathsf{cpabe}} = (\tilde{N}, X_3, g_q, g_q^{\tilde{a}}, e(g_q, g_q)^{\tilde{\alpha}}, \ \{U_{q,x} = g_q^{\tilde{u}_x}\}_{x \in \mathbb{S}})$ from the challenger, where $g_q \in \mathbb{G}_q$ and $X_3 \in \mathbb{G}_{p_3}$ are generators of subgroups $\mathbb{G}_q$ and $\mathbb{G}_{p_3}$ respectively, and $\tilde{a}, \tilde{\alpha}, \tilde{u}_x (x \in \mathbb{S}) \in \mathbb{Z}_{\tilde{N}}$ are random exponents.

$\mathcal{B}$ chooses generators $g_p, f_p \in \mathbb{G}_p$ and random exponents

$$\beta, \gamma, \delta \in \mathbb{Z}_N, \quad \{r_i, \ z_i, \ \alpha'_i \in \mathbb{Z}_N\}_{1 \le i \le m}, \quad \{c_j \in \mathbb{Z}_N\}_{1 \le j \le m}, \quad \{u_{p,x} \in \mathbb{Z}_p\}_{x \in \mathbb{S}}.$$

In addition, it randomly chooses $\theta \in \mathbb{Z}_N$. Then $\mathcal{B}$ gives $\mathcal{A}$ the following public parameters PK:

$$\begin{gathered} g = g_q g_p, \ h = g^{\theta}, \ f = g_q^{\tilde{a}} f_p, \ \ G = g^{\delta} g_p^{\gamma}, \ H = h^{\delta}, \ G_q = g_q^{\beta}, \ F_q = g_q^{\tilde{a}\beta}, \\ \{G_i = g^{r_i}, \ G_{q,i} = g_q^{\beta r_i}, \ H_i = g^{\theta r_i}, \ H_{q,i} = g_q^{\theta\beta r_i}, \ \hat{G}_i = g^{z_i}, \\ E_i = e(g_q, g_q)^{\tilde{\alpha}} e(g, g)^{\alpha'_i}, \ E_{q,i} = (e(g_q, g_q)^{\tilde{\alpha}} e(g_q, g_q)^{\alpha'_i})^{\beta} \ \}_{1 \le i \le m}, \\ \{\bar{G}_j = g^{c_j}\}_{1 \le j \le m}, \ \ \{U_x = U_{q,x} g_p^{u_{p,x}}\}_{x \in \mathbb{S}}. \end{gathered}$$

Note that $\mathcal{B}$ implicitly chooses $\alpha_1, \ldots, \alpha_m, u_x (\forall x \in \mathbb{S}) \in \mathbb{Z}_N$, $h_q, f_q \in \mathbb{G}_q$ and $h_p \in \mathbb{G}_p$ such that

$$\begin{gathered} \tilde{\alpha} + \alpha'_i \equiv \alpha_i \bmod q \ \forall i \in \{1, \ldots, m\}, \quad \alpha'_i \equiv \alpha_i \bmod p \ \forall i \in \{1, \ldots, m\}, \\ \tilde{u}_x \equiv u_x \bmod q \ \forall x \in \mathbb{S}, \quad u_{p,x} \equiv u_x \bmod p \ \forall x \in \mathbb{S}, \\ h_q = g_q^{\theta}, \ f_q = g_q^{\tilde{a}}, \quad h_p = g_p^{\theta}. \end{gathered}$$

[9] The situation is similar to that of the proof in [7,8] in the sense that the challenge is given in a subgroup of a composite group and the factors are given to the simulator.

[10] Note that in the original scheme of [21] X_3 is in the master secret key rather than in the public parameters, as X_3 is never used in encryption and decryption. Publishing X_3 in the public parameters will not affect the security of the scheme, as in the proof the simulator receives X_3 explicitly from the underlying assumptions (Assumption 1, 2, and 3) and can provide it to the adversary in the public parameters.

Phase 1. $\mathcal{A}$ issues adaptive private key queries. To respond to a query for $((i,j), S_{(i,j)})$, $\mathcal{B}$ submits $S_{(i,j)}$ to the challenger, and receives a decryption key

$$\mathsf{SK}^{\mathsf{cpabe}}_{S_{(i,j)}} = \big(\tilde{K} = g_q^{\tilde{\alpha}} g_q^{\tilde{a}\tilde{\sigma}} R,\ \tilde{K}' = g_q^{\tilde{\sigma}} R',\ \ \{\tilde{K}_x = U_{q,x}^{\tilde{\sigma}} R_x\}_{x \in S_{(i,j)}}\big),$$

where $\tilde{\sigma} \in \mathbb{Z}_{\tilde{N}}, R, R', R_x (x \in S_{(i,j)}) \in \mathbb{G}_{p_3}$ are randomly chosen and unknown to $\mathcal{B}$. $\mathcal{B}$ randomly chooses $\sigma_{p,i,j} \in \mathbb{Z}_p$ and $R'' \in \mathbb{G}_{p_3}$, then gives $\mathcal{A}$

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

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 X_3. The distribution of the private keys is same with that of the real scheme, where $\sigma_{i,j}$ is implicitly chosen such that $\tilde{\sigma} \equiv \sigma_{i,j} \bmod q$ and $\sigma_{p,i,j} \equiv \sigma_{i,j} \bmod p$.

Challenge. $\mathcal{A}$ submits to $\mathcal{B}$ an LSSS matrix (A^*, ρ) of size $L \times D$ 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

$$CT^{\mathsf{cpabe}} = \langle (A^*, \rho),\ \ \tilde{C} = M_b \cdot e(g_q, g_q)^{\tilde{\alpha}\tilde{\pi}},\ \tilde{C}_0 = g_q^{\tilde{\pi}},\ \ \{\tilde{C}_l = g_q^{\tilde{a} A_l^* \cdot \tilde{\boldsymbol{v}}} U_{q,\rho(l)}^{-\tilde{\xi}_l},\ \tilde{C}'_l = g_q^{\tilde{\xi}_l}\}_{l=1}^{L} \rangle,$$

where $\tilde{\boldsymbol{v}} = (\tilde{\pi}, \tilde{v}_2, \ldots, \tilde{v}_D) \in \mathbb{Z}_{\tilde{N}}^D$ and $\{\tilde{\xi}_l \in \mathbb{Z}_{\tilde{N}}\}_{l=1}^L$ are randomly chosen and unknown to $\mathcal{B}$. $\mathcal{B}$ first chooses random exponents

$$\begin{gathered}\kappa,\ \tau,\quad s'_1, \ldots, s'_m,\quad d_1, \ldots, d_m,\quad t_1, \ldots, t_m\ \in \mathbb{Z}_N, \\ (\xi'_1, \ldots, \xi'_L)\ \in \mathbb{Z}_N^L,\quad \boldsymbol{v}' = (\pi', v'_2, \ldots, v'_D)\ \in \mathbb{Z}_N^D,\end{gathered}$$

then creates challenge ciphertext $CT = \langle (A^* \rho),\ (R_i, \bar{R}_i, Q_i, \bar{Q}_i, \hat{Q}_i, T_i)_{i=1}^m, (C_j, \bar{C}_j)_{j=1}^m,\ (P_l, \bar{P}_l)_{l=1}^L \rangle$ for $(\bar{i} = 1, \bar{j} = 1)$ as follows:

1. For each $i \in \{1, \ldots, m\}$:
   - if $i > 1$:

     $$R_i = G_{q,i}^{s'_i} \tilde{C}_0^{r_i/\tau},\ \ \bar{R}_i = H_{q,i}^{\kappa s'_i} \tilde{C}_0^{\theta\kappa r_i/\tau},\quad Q_i = G_q^{\tau s'_i} \tilde{C}_0,\ \ \bar{Q}_i = F_q^{\tau s'_i} \hat{G}_i^{d_i} f^{\pi'},\ \ \hat{Q}_i = g^{d_i},$$
     $$T_i = \tilde{C} \cdot e(g_q^{\alpha'_i}, \tilde{C}_0) \cdot E_{q,i}^{\tau s'_i}.$$

   - if $i = 1$:

     $$R_i = G_i^{s'_i} \tilde{C}_0^{r_i/\tau},\ \ \bar{R}_i = H_i^{\kappa s'_i} \tilde{C}_0^{\theta\kappa r_i/\tau},\quad Q_i = g^{\tau s'_i} \tilde{C}_0,\ \ \bar{Q}_i = f^{\tau s'_i} \hat{G}_i^{d_i} f^{\pi'},\ \ \hat{Q}_i = g^{d_i},$$
     $$T_i = \tilde{C} \cdot e(g_q^{\alpha'_i}, \tilde{C}_0) \cdot E_i^{\tau s'_i}.$$

2. For each $j \in \{1, \ldots, m\}$: $C_j = \bar{G}_j^{\tau} h^{\kappa t_j},\ \ \bar{C}_j = g^{t_j}$.
3. For each $l \in \{1, \ldots, L\}$: $P_l = f^{A_l^* \cdot \boldsymbol{v}'} U_{\rho(l)}^{-\xi'_l} / \tilde{C}_l,\ \ \bar{P}_l = g^{\xi'_l} / \tilde{C}'_l$.

Note that $\mathcal{B}$ implicitly chooses $s_1, \ldots, s_m \in \mathbb{Z}_N$, $(\xi_1, \ldots, \xi_L) \in \mathbb{Z}_N^L$, and $\boldsymbol{v} = (\pi, v_2, \ldots, v_D) \in \mathbb{Z}_N^D$ such that

$$\begin{aligned} \pi' - \tilde{\pi} \equiv \pi \bmod q, &\quad \pi' \equiv \pi \bmod p, \\ s'_1 + \tilde{\pi}/\tau \equiv s_1 \bmod q, &\quad s'_1 \equiv s_1 \bmod p, \\ \forall i \in \{2, \ldots, m\} :\ s'_i + \frac{\tilde{\pi}}{\beta\tau} \equiv s_i \bmod q, &\quad s'_i \equiv s_i \bmod p, \\ \forall l \in \{1, \ldots, L\} :\ \xi'_l - \tilde{\xi}_l \equiv \xi_l \bmod q, &\quad \xi'_l \equiv \xi_l \bmod p, \\ \forall d \in \{2, \ldots, D\} :\ v'_d - \tilde{v}_d \equiv v_d \bmod q, &\quad v'_d \equiv v_d \bmod p. \end{aligned}$$

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

D.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 simulator $\mathcal{B}$ to solve a Decision (Modified) 3-party Diffie-Hellman problem instance as follows.

$\mathcal{B}$ receives the Decision (Modified) 3-party Diffie-Hellman challenge from the challenger as

$$(g_p, A = g_p^a, B = g_p^b, C = g_p^c, D = g_p^{b^2}, T).$$

The challenge will be given in the subgroup $\mathbb{G}_p$ of prime order p in a composite group $\mathbb{G}_N$ of order $N = qpp_2p_3$. $\mathcal{B}$ is given the factors q, p, p_2, p_3. Since the game is played in the subgroup $\mathbb{G}_p$, $\mathcal{B}$ chooses for itself everything in the subgroups $\mathbb{G}_q$ and $\mathbb{G}_{p_3}$ in the following simulation.

At this point $\mathcal{B}$ does not know which case $\mathcal{A}$ will behave in, so $\mathcal{B}$ will guess it by flipping a random coin $\tilde{b} \in \{0, 1\}$. We will show that in both cases $\mathcal{B}$'s output will be the same as that in the real game, i.e., the value of $\tilde{b}$ is hidden from $\mathcal{A}$. Then $\mathcal{B}$ can finish the simulation with probability $1/2$. If $\tilde{b} = 0$, $\mathcal{B}$ will behave in the following **Case I**, otherwise **Case II**.

Case I:

Setup. $\mathcal{B}$ chooses generators $g_q, h_q, f_q \in \mathbb{G}_q, X_3 \in \mathbb{G}_{p_3}$, and random exponents

$$\beta, \gamma, \delta \in \mathbb{Z}_N, \quad \{r_{q,i} \in \mathbb{Z}_q,\ r'_{p,i} \in \mathbb{Z}_p,\ z_i, \alpha_i \in \mathbb{Z}_N\}_{1 \le i \le m},$$
$$\{c_{q,j} \in \mathbb{Z}_q,\ c'_{p,j} \in \mathbb{Z}_p\}_{1 \le j \le m}, \quad \{u_x \in \mathbb{Z}_N\}_{x \in \mathbb{S}}.$$

In addition, it randomly chooses $\eta \in \mathbb{Z}_p$. Then $\mathcal{B}$ gives $\mathcal{A}$ the following public parameters PK:

$$g = g_q g_p,\ h = h_q B,\ f = f_q g_p^{\eta},\ \ G = g^{\delta} g_p^{\gamma},\ H = h^{\delta},\ G_q = g_q^{\beta},\ F_q = f_q^{\beta},$$
$$\{G_i = g_q^{r_{q,i}} g_p^{r'_{p,i}},\ H_i = h_q^{r_{q,i}} B^{r'_{p,i}}\}_{1 \le i \le m, i \ne \bar{i}},\ \ G_{\bar{i}} = g_q^{r_{q,\bar{i}}} B^{r'_{p,\bar{i}}},\ H_{\bar{i}} = h_q^{r_{q,\bar{i}}} D^{r'_{p,\bar{i}}},$$
$$\{G_{q,i} = g_q^{\beta r_{q,i}},\ H_{q,i} = h_q^{\beta r_{q,i}},\ \hat{G}_i = g^{z_i},\ \ E_i = e(g, g)^{\alpha_i},\ E_{q,i} = e(g_q, g_q)^{\beta \alpha_i}\}_{1 \le i \le m},$$
$$\{\bar{G}_j = g_q^{c_{q,j}} g_p^{c'_{p,j}}\}_{1 \le j \le m, j \ne \bar{j}},\ \bar{G}_{\bar{j}} = g_q^{c_{q,\bar{j}}} C^{c'_{p,\bar{j}}},\ \ \{U_x = g^{u_x}\}_{x \in \mathbb{S}}.$$

Note that $\mathcal{B}$ implicitly chooses $r_1, \ldots, r_m,\ c_1, \ldots, c_m\ \in \mathbb{Z}_N$ and $h_p, f_p \in \mathbb{G}_p$ such that

$$r_{q,i} \equiv r_i \bmod q\ \forall i \in \{1, \ldots, m\}, \quad r'_{p,i} \equiv r_i \bmod p\ \forall i \in \{1, \ldots, m\} \setminus \{\bar{i}\}, \quad br'_{p,\bar{i}} \equiv r_{\bar{i}} \bmod p,$$
$$c_{q,j} \equiv c_j \bmod q\ \forall j \in \{1, \ldots, m\}, \quad c'_{p,j} \equiv c_j \bmod p\ \forall j \in \{1, \ldots, m\} \setminus \{\bar{j}\}, \quad cc'_{p,\bar{j}} \equiv c_{\bar{j}} \bmod p,$$
$$h_p = B, \quad f_p = g_p^{\eta}.$$

Key Query. $\mathcal{A}$ issues adaptive private key queries. To respond to a query for $((i, j), S_{(i,j)})$, if $(i, j) = (\bar{i}, \bar{j})$, $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ randomly chooses $\sigma_{i,j} \in \mathbb{Z}_N$ and $R, R', R'', R_x(x \in$

$S_{(i,j)}) \in \mathbb{G}_{p_3}$, and creates the private key

$$\begin{aligned}
&\mathsf{SK}_{(i,j),S_{(i,j)}} = \big(K_{i,j},\ K'_{i,j},\ K''_{i,j},\ \{K_{i,j,x}\}_{x\in S_{(i,j)}}\big)\\
&= \begin{cases}
\big(g^{\alpha_i} g_q^{r_{q,i}c_{q,j}} g_p^{r'_{p,i}c'_{p,j}} f^{\sigma_{i,j}}R,\ g^{\sigma_{i,j}}R',\ g^{z_i\sigma_{i,j}}R'',\ \ \{U_x^{\sigma_{i,j}}R_x\}_{x\in S_{(i,j)}}\big), & : i \neq \bar{i}, j \neq \bar{j}\\
\big(g^{\alpha_i} g_q^{r_{q,i}c_{q,j}} B^{r'_{p,i}c'_{p,j}} f^{\sigma_{i,j}}R,\ g^{\sigma_{i,j}}R',\ g^{z_i\sigma_{i,j}}R'',\ \ \{U_x^{\sigma_{i,j}}R_x\}_{x\in S_{(i,j)}}\big), & : i = \bar{i}, j \neq \bar{j}\\
\big(g^{\alpha_i} g_q^{r_{q,i}c_{q,j}} C^{r'_{p,i}c'_{p,j}} f^{\sigma_{i,j}}R,\ g^{\sigma_{i,j}}R',\ g^{z_i\sigma_{i,j}}R'',\ \ \{U_x^{\sigma_{i,j}}R_x\}_{x\in S_{(i,j)}}\big). & : i \neq \bar{i}, j = \bar{j}
\end{cases}
\end{aligned}$$

Challenge. $\mathcal{A}$ submits a message M and an attribute set S^*. $\mathcal{B}$ constructs the LSSS matrix (A,ρ) for $\mathbb{A}_{S^*}$. Let $L \times D$ be the size of (A,ρ), $\mathcal{B}$ chooses random exponents

$$\begin{aligned}
\kappa,\quad d_1,\dots,d_m,\quad t'_1,\dots,t'_m\ &\in \mathbb{Z}_N,\\
\tau_q,\quad s_{q,\bar{i}},\dots,s_{q,m},\quad \mu_{q,1},\dots,\mu_{q,\bar{j}-1}\ &\in \mathbb{Z}_q,\\
s'_{p,\bar{i}},\qquad\qquad \mu'_{p,1},\dots,\mu'_{p,\bar{j}-1}\ &\in \mathbb{Z}_p,\\
(\nu_{1,1},\nu_{1,2},\nu_{1,3}),\dots,(\nu_{\bar{i}-1,1},\nu_{\bar{i}-1,2},\nu_{\bar{i}-1,3})\ &\in \mathbb{Z}_N^3,\\
(\xi_1,\dots,\xi_L)\ \in \mathbb{Z}_N^L,\quad \boldsymbol{v} = (\pi, v_2,\dots,v_D)\ &\in \mathbb{Z}_N^D.
\end{aligned}$$

Then $\mathcal{B}$ creates the ciphertext $\langle (A,\rho),\ (R_i,\bar{R}_i,Q_i,\bar{Q}_i,\hat{Q}_i,T_i)_{i=1}^m,\ \ (C_j,\bar{C}_j)_{j=1}^m,\ \ (P_l,\bar{P}_l)_{l=1}^L\rangle$ as follows:

1. For each $i \in \{1,\dots,m\}$:
   - if $i > \bar{i}$:
   $$\begin{aligned}
   &R_i = g_q^{\beta r_{q,i}s_{q,i}},\quad \bar{R}_i = h_q^{\beta r_{q,i}\kappa s_{q,i}},\quad Q_i = g_q^{\beta\tau_q s_{q,i}},\\
   &\bar{Q}_i = f_q^{\beta\tau_q s_{q,i}} g^{z_i d_i} f^{\pi},\quad \hat{Q}_i = g^{d_i},\quad T_i = M\cdot e(g_q,g_q)^{\beta\alpha_i\tau_q s_{q,i}}.
   \end{aligned}$$
   - if $i = \bar{i}$:
   $$\begin{aligned}
   &R_i = g_q^{r_{q,i}s_{q,i}} g_p^{r'_{p,i}s'_{p,\bar{i}}},\quad \bar{R}_i = h_q^{r_{q,i}\kappa s_{q,i}} B^{r'_{p,\bar{i}}\kappa s'_{p,\bar{i}}},\quad Q_i = g_q^{\tau_q s_{q,i}} A^{s'_{p,\bar{i}}},\\
   &\bar{Q}_i = f_q^{\tau_q s_{q,i}} A^{\eta s'_{p,\bar{i}}} g^{z_i d_i} f^{\pi},\quad \hat{Q}_i = g^{d_i},\quad T_i = M\cdot e(g_q,g_q)^{\alpha_i\tau_q s_{q,i}}\cdot e(g_p,A)^{\alpha_i s'_{p,\bar{i}}}.
   \end{aligned}$$
   - if $i < \bar{i}$:
   $$R_i = g^{\nu_{i,1}},\quad \bar{R}_i = h^{\kappa\nu_{i,1}},\quad Q_i = g^{\nu_{i,2}},\quad \bar{Q}_i = f^{\nu_{i,2}} g^{z_i d_i} f^{\pi},\quad \hat{Q}_i = g^{d_i},\quad T_i = e(g,g)^{\nu_{i,3}}.$$
2. For each $j \in \{1,\dots,m\}$:
   - if $j > \bar{j}$: $C_j = g_q^{c_{q,j}\tau_q} h^{\kappa t'_j}$, $\qquad \bar{C}_j = A^{-c'_{p,j}/\kappa} g^{t'_j}$.
   - if $j = \bar{j}$: $C_j = g_q^{c_{q,j}\tau_q} T^{c'_{p,j}} h^{\kappa t'_j}$, $\bar{C}_j = g^{t'_j}$.
   - if $j < \bar{j}$: $C_j = g_q^{c_{q,j}\tau_q} g_p^{\mu'_{p,j}} h^{\kappa t'_j}$, $\bar{C}_j = g^{t'_j}$.
3. For each $l \in \{1,\dots,L\}$: $P_l = f^{A_l\cdot\boldsymbol{v}} U_{\rho(l)}^{-\xi_l}$, $\bar{P}_l = g^{\xi_l}$.

Note that $\mathcal{B}$ implicitly chooses $\tau,\ s_{\bar{i}},\dots,s_m,\ t_1,\dots,t_m,\ \mu_1,\dots,\mu_{\bar{j}-1}\ \in \mathbb{Z}_N$ such that

$$\begin{aligned}
&\tau_q \equiv \tau \bmod q,\quad a\cdot b \equiv \tau \bmod p,\\
&s_{q,i} \equiv s_i \bmod q\ \forall i \in \{\bar{i},\dots,m\},\quad s'_{p,\bar{i}}/b \equiv s_{\bar{i}} \bmod p,\\
&\forall j \in \{1,\dots,\bar{j}-1\}:\\
&\qquad t'_j - \delta\mu_{q,j} \equiv t_j \bmod q,\quad t'_j - (\delta+\gamma)(c'_{p,j}ab - \mu'_{p,j})/(b\kappa\gamma) \equiv t_j \bmod p,\\
&\qquad \mu_{q,j} \equiv \mu_j \bmod q,\quad (c'_{p,j}ab - \mu'_{p,j})/(b\kappa\gamma) \equiv \mu_j \bmod p,\\
&\forall j \in \{\bar{j}+1,\dots,m\}:\\
&\qquad t'_j \equiv t_j \bmod q,\quad t'_j - ac'_{p,j}/\kappa \equiv t_j \bmod p.
\end{aligned}$$

If $T = g_p^{abc}$, then the ciphertext is a well-formed encryption to the index $(\bar{i}, \bar{j})$ with implicitly setting $t_{\bar{j}} = t'_{\bar{j}}$. If T is randomly chosen, say $T = g_p^r$ for some random $r \in \mathbb{Z}_p$, the ciphertext is a well-formed encryption to the index $(\bar{i}, \bar{j}+1)$ with implicitly setting $t_{\bar{j}}$ and $\mu_{\bar{j}}$ such that

$$\begin{aligned} t'_{\bar{j}} - \delta\mu_{q,\bar{j}} \equiv t_{\bar{j}} \bmod q, &\quad t'_{\bar{j}} - (\delta+\gamma)c'_{p,\bar{j}}(abc - r)/(b\kappa\gamma) \equiv t_{\bar{j}} \bmod p, \\ \mu_{q,\bar{j}} \equiv \mu_{\bar{j}} \bmod q, &\quad c'_{p,\bar{j}}(abc - r)/(b\kappa\gamma) \equiv \mu_{\bar{j}} \bmod p, \end{aligned}$$

for some random $\mu_{q,\bar{j}} \in \mathbb{Z}_q$.

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 the Decision (Modified) 3-party Diffie-Hellman game.

Note that the distributions of the public parameters, private keys and challenge ciphertext are same as the real scheme, $\mathcal{B}$'s advantage in the Decision (Modified) 3-party Diffie-Hellman game will be exactly equal to $\mathcal{A}$'s advantage in the Index Hiding Game.

Case II:

Setup. Firstly, $\mathcal{B}$ randomly chooses an attribute $\bar{x} \in \mathbb{S}$ to guess that $\bar{x}$ will be in $S^* \setminus S_{(\bar{i},\bar{j})}$. Then $\mathcal{B}$ chooses generators $g_q, h_q, f_q \in \mathbb{G}_q, X_3 \in \mathbb{G}_{p_3}$, and random exponents

$$\begin{aligned} &\beta, \gamma, \delta \in \mathbb{Z}_N, \quad \{r_{q,i},\ z_{q,i} \in \mathbb{Z}_q, \quad r'_{p,i},\ z'_{p,i} \in \mathbb{Z}_p, \quad \alpha_i \in \mathbb{Z}_N\}_{1 \le i \le m}, \quad \{c_{q,j} \in \mathbb{Z}_q,\ c'_{p,j} \in \mathbb{Z}_p\}_{1 \le j \le m}, \\ &\{u_x \in \mathbb{Z}_N\}_{x \in \mathbb{S} \setminus \{\bar{x}\}}, \quad u_{q,\bar{x}} \in \mathbb{Z}_q, \quad u'_{p,\bar{x}} \in \mathbb{Z}_p. \end{aligned}$$

In addition, it randomly chooses $\eta \in \mathbb{Z}_p$. $\mathcal{B}$ gives $\mathcal{A}$ the following public parameters PK:

$$\begin{aligned} &g = g_q g_p,\ h = h_q B,\ f = f_q B^{r'_{p,\bar{i}}} g_p^{\eta},\ \ G = g^{\delta} g_p^{\gamma},\ H = h^{\delta},\ G_q = g_q^{\beta},\ F_q = f_q^{\beta}, \\ &\{G_i = g_q^{r_{q,i}} g_p^{r'_{p,i}},\ H_i = h_q^{r_{q,i}} B^{r'_{p,i}},\ \hat{G}_i = g_q^{z_{q,i}} B^{z'_{p,i}}\}_{1 \le i \le m, i \ne \bar{i}}, \\ &\ \ G_{\bar{i}} = g_q^{r_{q,\bar{i}}} B^{r'_{p,\bar{i}}},\ H_{\bar{i}} = h_q^{r_{q,\bar{i}}} D^{r'_{p,\bar{i}}}, \hat{G}_{\bar{i}} = g_q^{z_{q,\bar{i}}} g_p^{z'_{p,\bar{i}}}, \\ &\{G_{q,i} = g_q^{\beta r_{q,i}},\ H_{q,i} = h_q^{\beta r_{q,i}}, \quad E_i = e(g,g)^{\alpha_i},\ E_{q,i} = e(g_q, g_q)^{\beta\alpha_i}\}_{1 \le i \le m}, \\ &\{\bar{G}_j = g_q^{c_{q,j}} g_p^{c'_{p,j}}\}_{1 \le j \le m, j \ne \bar{j}}, \quad \bar{G}_{\bar{j}} = g_q^{c_{q,\bar{j}}} C^{c'_{p,\bar{j}}}, \\ &\{U_x = g^{u_x}\}_{x \in \mathbb{S} \setminus \{\bar{x}\}}, \quad U_{\bar{x}} = g_q^{u_{q,\bar{x}}} B^{u'_{p,\bar{x}}}. \end{aligned}$$

Note that $\mathcal{B}$ implicitly chooses $r_1, \ldots, r_m,\ c_1, \ldots, c_m,\ z_1, \ldots, z_m,\ u_{\bar{x}}\ \in \mathbb{Z}_N$ and $h_p, f_p \in \mathbb{G}_p$ such that

$$\begin{aligned} &r_{q,i} \equiv r_i \bmod q\ \forall i \in \{1, \ldots, m\}, \quad r'_{p,i} \equiv r_i \bmod p\ \forall i \in \{1, \ldots, m\} \setminus \{\bar{i}\}, \quad br'_{p,\bar{i}} \equiv r_{\bar{i}} \bmod p, \\ &c_{q,j} \equiv c_j \bmod q\ \forall j \in \{1, \ldots, m\}, \quad c'_{p,j} \equiv c_j \bmod p\ \forall j \in \{1, \ldots, m\} \setminus \{\bar{j}\}, \quad cc'_{p,\bar{j}} \equiv c_{\bar{j}} \bmod p, \\ &z_{q,i} \equiv z_i \bmod q\ \forall i \in \{1, \ldots, m\}, \quad bz'_{p,i} \equiv z_i \bmod p\ \forall i \in \{1, \ldots, m\} \setminus \{\bar{i}\}, \quad z'_{p,\bar{i}} \equiv z_{\bar{i}} \bmod p, \\ &u_{q,\bar{x}} \equiv u_{\bar{x}} \bmod q, \quad bu'_{p,\bar{x}} \equiv u_{\bar{x}} \bmod p, \quad h_p = B, \quad f_p = B^{r'_{p,\bar{i}}} g_p^{\eta}. \end{aligned}$$

Key Query. $\mathcal{A}$ issues adaptive private key queries. To respond to a query for $((i,j), S_{(i,j)})$, if $(i,j) \ne (\bar{i}, \bar{j})$, $\mathcal{B}$ randomly chooses $\sigma_{i,j} \in \mathbb{Z}_N$. Otherwise (i.e., $(i,j) = (\bar{i}, \bar{j})$), if $\bar{x} \in S_{(i,j)}$ then $\mathcal{B}$ aborts, otherwise $\mathcal{B}$ randomly chooses $\sigma_{q,i,j} \in \mathbb{Z}_q, \sigma'_{p,i,j} \in \mathbb{Z}_p$ and sets the value of $\sigma_{i,j}$ by implicitly setting $\sigma_{q,i,j} \equiv \sigma_{i,j} \bmod q, \sigma'_{p,i,j} - cc'_{p,j} \equiv \sigma_{i,j} \bmod p$. In addition $\mathcal{B}$ randomly chooses $R, R', R'', R_x (x \in S_{(i,j)}) \in \mathbb{G}_{p_3}$. $\mathcal{B}$ creates the private key

$$\begin{aligned}\mathsf{SK}_{(i,j),S_{(i,j)}} &= \big(K_{i,j},\ K'_{i,j},\ K''_{i,j},\ \{K_{i,j,x}\}_{x\in S_{(i,j)}}\big)\\ &= \begin{cases} \big(g^{\alpha_i} g_q^{r_{q,i}c_{q,j}} g_p^{r'_{p,i}c'_{p,j}} f^{\sigma_{i,j}} R,\ g^{\sigma_{i,j}} R',\ \hat{G}_i^{\sigma_{i,j}} R'',\ \{U_x^{\sigma_{i,j}} R_x\}_{x\in S_{(i,j)}}\big), & : i \neq \bar{i}, j \neq \bar{j} \\ \big(g^{\alpha_i} g_q^{r_{q,i}c_{q,j}} B^{r'_{p,i}c'_{p,j}} f^{\sigma_{i,j}} R,\ g^{\sigma_{i,j}} R',\ \hat{G}_i^{\sigma_{i,j}} R'',\ \{U_x^{\sigma_{i,j}} R_x\}_{x\in S_{(i,j)}}\big), & : i = \bar{i}, j \neq \bar{j} \\ \big(g^{\alpha_i} g_q^{r_{q,i}c_{q,j}} C^{r'_{p,i}c'_{p,j}} f^{\sigma_{i,j}} R,\ g^{\sigma_{i,j}} R',\ \hat{G}_i^{\sigma_{i,j}} R'',\ \{U_x^{\sigma_{i,j}} R_x\}_{x\in S_{(i,j)}}\big), & : i \neq \bar{i}, j = \bar{j} \\ \big(g^{\alpha_i} g_q^{r_{q,i}c_{q,j}} f_q^{\sigma_{q,i,j}} (B^{r'_{p,\bar{i}}} g_p^{\eta})^{\sigma'_{p,i,j}} C^{-\eta c'_{p,j}} R,\ g_q^{\sigma_{q,i,j}} g_p^{\sigma'_{p,i,j}} C^{-c'_{p,j}} R', & \\ \quad g_q^{z_{q,i}\sigma_{q,i,j}} (g_p^{\sigma'_{p,i,j}} C^{-c'_{p,j}})^{z'_{p,i}} R'',\ \{(g_q^{\sigma_{q,i,j}} g_p^{\sigma'_{p,i,j}} C^{-c'_{p,j}})^{u_x} R_x\}_{x\in S_{(i,j)}}\big). & : i = \bar{i}, j = \bar{j} \end{cases}\end{aligned}$$

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

$$\begin{array}{rr} \kappa,\ \ t'_1, \ldots, t'_m & \in \mathbb{Z}_N, \\ \tau_q,\ \ s_{q,\bar{i}}, \ldots, s_{q,m},\ \ d_{q,1}, \ldots, d_{q,m},\ \ \mu_{q,1}, \ldots, \mu_{q,\bar{j}-1} & \in \mathbb{Z}_q, \\ s'_{p,\bar{i}},\qquad\qquad d'_{p,1}, \ldots, d'_{p,m},\ \ \mu'_{p,1}, \ldots, \mu'_{p,\bar{j}-1} & \in \mathbb{Z}_p, \\ (\nu_{1,1}, \nu_{1,2}, \nu_{1,3}), \ldots, (\nu_{\bar{i}-1,1}, \nu_{\bar{i}-1,2}, \nu_{\bar{i}-1,3}) & \in \mathbb{Z}_N^3, \\ \{\xi_{q,l}\ \in \mathbb{Z}_q,\ \xi'_{p,l}\ \in \mathbb{Z}_p\}_{\forall l \in \{1,\ldots,L\}\ s.t.\ \rho(l)=\bar{x}},\qquad \pi_q\ \in \mathbb{Z}_q,\ \pi'_p & \in \mathbb{Z}_p, \\ \{\xi_l\ \in \mathbb{Z}_N\}_{\forall l \in \{1,\ldots,L\}\ s.t.\ \rho(l)\neq\bar{x}},\qquad \boldsymbol{v}' & \in \mathbb{Z}_N^D, \end{array}$$

with the first entry of $\boldsymbol{v}'$ equal to zero.
Implicitly setting

$$\begin{aligned} &\pi_q \equiv \pi \bmod q,\quad \pi'_p - as'_{p,\bar{i}} \equiv \pi \bmod p,\quad \boldsymbol{v} = \pi\boldsymbol{w} + \boldsymbol{v}', \\ &\xi_{q,l} \equiv \xi_l \bmod q,\quad \xi'_{p,l} - ar'_{p,\bar{i}}s'_{p,\bar{i}}(A_l \cdot \boldsymbol{w})/u'_{p,\bar{x}} \equiv \xi_l \bmod p \quad \forall l \in \{1, \ldots, L\}\ s.t.\ \rho(l) = \bar{x}, \end{aligned}$$

$\mathcal{B}$ creates the ciphertext $\langle (A, \rho),\ (R_i, \bar{R}_i, Q_i, \bar{Q}_i, \hat{Q}_i, T_i)_{i=1}^m,\ (C_j, \bar{C}_j)_{j=1}^m,\ (P_l, \bar{P}_l)_{l=1}^L \rangle$ as follows:

1. For each $i \in \{1, \ldots, m\}$:
   - if $i > \bar{i}$:

     $$R_i = g_q^{\beta r_{q,i} s_{q,i}},\quad \bar{R}_i = h_q^{\beta r_{q,i} \kappa s_{q,i}},$$

     $$Q_i = g_q^{\beta \tau_q s_{q,i}},\quad \bar{Q}_i = f_q^{\beta \tau_q s_{q,i} + \pi_q} g_q^{z_{q,i} d_{q,i}} (B^{r'_{p,\bar{i}}} g_p^{\eta})^{\pi'_p} A^{-\eta s'_{p,\bar{i}}} B^{z'_{p,i} d'_{p,i}},\quad \hat{Q}_i = g_q^{d_{q,i}} g_p^{d'_{p,i}} A^{\frac{r'_{p,\bar{i}} s'_{p,\bar{i}}}{z'_{p,i}}},$$
     $$T_i = M \cdot e(g_q, g_q)^{\beta \alpha_i \tau_q s_{q,i}}.$$

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

     $$R_i = g_q^{r_{q,i} s_{q,i}} g_p^{r'_{p,\bar{i}} s'_{p,\bar{i}}},\quad \bar{R}_i = h_q^{r_{q,i} \kappa s_{q,i}} B^{r'_{p,\bar{i}} \kappa s'_{p,\bar{i}}},$$
     $$Q_i = g_q^{\tau_q s_{q,i}} A^{s'_{p,\bar{i}}},\quad \bar{Q}_i = f_q^{\tau_q s_{q,i} + \pi_q} (B^{r'_{p,\bar{i}}} g_p^{\eta})^{\pi'_p} g_q^{z_{q,\bar{i}} d_{q,\bar{i}}} g_p^{z'_{p,\bar{i}} d'_{p,\bar{i}}},\quad \hat{Q}_i = g_q^{d_{q,\bar{i}}} g_p^{d'_{p,\bar{i}}},$$
     $$T_i = M \cdot e(g_q, g_q)^{\alpha_i \tau_q s_{q,i}} \cdot e(g_p, A)^{\alpha_i s'_{p,\bar{i}}}.$$

- if $i < \bar{i}$:

  $$R_i = g^{\nu_{i,1}}, \quad \bar{R}_i = h^{\kappa \nu_{i,1}},$$

  $$Q_i = g^{\nu_{i,2}}, \quad \bar{Q}_i = f^{\nu_{i,2}} f_q^{\pi_q} g_q^{z_{q,i} d_{q,i}} (B^{r'_{p,\bar{i}}} g_p^{\eta})^{\pi'_p} A^{-\eta s'_{p,\bar{i}}} B^{z'_{p,i} d'_{p,i}}, \quad \hat{Q}_i = g_q^{d_{q,i}} g_p^{d'_{p,i}} A^{r'_{p,\bar{i}} s'_{p,\bar{i}} / z'_{p,i}},$$

  $$T_i = e(g,g)^{\nu_{i,3}}.$$

2. For each $j \in \{1, \ldots, m\}$:
   - if $j > \bar{j}$: $C_j = g_q^{c_{q,j} \tau_q} h^{\kappa t'_j}$, $\quad \bar{C}_j = A^{-c'_{p,j}/\kappa} g^{t'_j}$.
   - if $j = \bar{j}$: $C_j = g_q^{c_{q,j} \tau_q} T^{c'_{p,j}} h^{\kappa t'_j}$, $\quad \bar{C}_j = g^{t'_j}$.
   - if $j < \bar{j}$: $C_j = g_q^{c_{q,j} \tau_q} g_p^{\mu'_{p,j}} h^{\kappa t'_j}$, $\quad \bar{C}_j = g^{t'_j}$.
3. For each $l \in \{1, \ldots, L\}$:
   - if $\rho(l) \neq \bar{x}$: since $A_l \cdot \boldsymbol{w} = 0$, we have $A_l \cdot \boldsymbol{v} = A_l \cdot (\pi \boldsymbol{w} + \boldsymbol{v}') = A_l \cdot \boldsymbol{v}'$, then

     $$P_l = f^{A_l \cdot \boldsymbol{v}'} U_{\rho(l)}^{-\xi_l}, \quad \bar{P}_l = g^{\xi_l}.$$

   - if $\rho(l) = \bar{x}$:

     $$\begin{aligned} P_l =& f_q^{\pi_q (A_l \cdot \boldsymbol{w})} (B^{r'_{p,\bar{i}}} g_p^{\eta})^{\pi'_p (A_l \cdot \boldsymbol{w})} A^{-\eta s'_{p,\bar{i}} (A_l \cdot \boldsymbol{w})} f^{A_l \cdot \boldsymbol{v}'} g_q^{-u_{q,\bar{x}} \xi_{q,l}} B^{-u'_{p,\bar{x}} \xi'_{p,l}}, \\ \bar{P}_l =& g_q^{\xi_{q,l}} g_p^{\xi'_{p,l}} A^{-r'_{p,\bar{i}} s'_{p,\bar{i}} (A_l \cdot \boldsymbol{w}) / u'_{p,\bar{x}}}. \end{aligned}$$

Note that $\mathcal{B}$ implicitly chooses $\tau,\ s_{\bar{i}}, \ldots, s_m, t_1, \ldots, t_m, d_1, \ldots, d_m, \mu_1, \ldots, \mu_{\bar{j}-1} \ \in \mathbb{Z}_N$ such that

$$\begin{aligned} \tau_q \equiv \tau \bmod q, &\quad a \cdot b \equiv \tau \bmod p, \\ s_{q,i} \equiv s_i \bmod q \ \forall i \in \{\bar{i}, \ldots, m\}, &\quad s'_{p,\bar{i}}/b \equiv s_{\bar{i}} \bmod p, \\ d_{q,i} \equiv d_i \bmod q \ \forall i \in \{1, \ldots, m\}, &\quad d'_{p,i} + a r'_{p,\bar{i}} s'_{p,\bar{i}} / z'_{p,i} \equiv d_i \bmod p \ \forall i \in \{1, \ldots, m\} \setminus \{\bar{i}\}, \\ &\quad d'_{p,\bar{i}} \equiv d_{\bar{i}} \bmod p, \\ \forall j \in \{1, \ldots, \bar{j}-1\} : & \\ t'_j - \delta \mu_{q,j} \equiv t_j \bmod q, &\quad t'_j - (\delta + \gamma)(c'_{p,j} ab - \mu'_{p,j})/(b \kappa \gamma) \equiv t_j \bmod p, \\ \mu_{q,j} \equiv \mu_j \bmod q, &\quad (c'_{p,j} ab - \mu'_{p,j})/(b \kappa \gamma) \equiv \mu_j \bmod p, \\ \forall j \in \{\bar{j}+1, \ldots, m\} : & \\ t'_j \equiv t_j \bmod q, &\quad t'_j - a c'_{p,j}/\kappa \equiv t_j \bmod p. \end{aligned}$$

If $T = g_p^{abc}$, then the ciphertext is a well-formed encryption to the index $(\bar{i}, \bar{j})$ with implicitly setting $t_{\bar{j}} = t'_{\bar{j}}$. If T is randomly chosen, say $T = g_p^r$ for some random $r \in \mathbb{Z}_p$, the ciphertext is a well-formed encryption to the index $(\bar{i}, \bar{j}+1)$ with implicitly setting $t_{\bar{j}}$ and $\mu_{\bar{j}}$ such that

$$\begin{aligned} t'_{\bar{j}} - \delta \mu_{q,\bar{j}} \equiv t_{\bar{j}} \bmod q, &\quad t'_{\bar{j}} - (\delta + \gamma) c'_{p,\bar{j}} (abc - r)/(b \kappa \gamma) \equiv t_{\bar{j}} \bmod p, \\ \mu_{q,\bar{j}} \equiv \mu_{\bar{j}} \bmod q, &\quad c'_{p,\bar{j}} (abc - r)/(b \kappa \gamma) \equiv \mu_{\bar{j}} \bmod p, \end{aligned}$$

for some random $\mu_{q,\bar{j}} \in \mathbb{Z}_q$.

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 the (Modified) 3-party Diffie-Hellman game.

Note that $S^* \setminus S_{(\bar{i},\bar{j})} \neq \emptyset$, we have that $\mathcal{B}$ can guess a right $\bar{x}$ and finish the simulation with probability $|S^* \setminus S_{(\bar{i},\bar{j})}|/|\mathbb{S}|$, which is at least $1/|\mathbb{S}|$. Also note that the distributions of the public parameters, private keys and challenge ciphertext are same as the real scheme, $\mathcal{B}$'s advantage in the (Modified) 3-party Diffie-Hellman game will be at least $\epsilon/|\mathbb{S}|$.

D.3 Proof of Claim 2

Boneh and Waters [8] proposed an AugBE scheme $\Sigma_{\mathsf{AugBE}} = (\mathsf{Setup}_{\mathsf{AugBE}}, \mathsf{Encrypt}_{\mathsf{AugBE}}, \mathsf{Decrypt}_{\mathsf{AugBE}})$ and proved that Σ_{AugBE} has index hiding property. In their proof of Lemma 5.3 in [8], two hybrid experiments

- H_2^{AugBE}: Encrypt to $(\bar{i}, m+1)$, (corresponding to H_2 in [8])
- H_3^{AugBE}: Encrypt to $(\bar{i}+1, 1)$, (corresponding to H_5 in [8])

were defined and proved indistinguishable by a sequence of hybrid sub-experiments and corresponding claims (Claim 5.5, 5.6 and 5.7 in [8]). Our Claim 2 can be proved using similar proof, but due to the space restriction and for simplicity, we prove Claim 2 by a reduction from the indistinguishability of H_1 and H_2 for our AugCP-ABE scheme Σ_{A} to that of H_2^{AugBE} and H_3^{AugBE} for the AugBE scheme Σ_{AugBE} in [8]. Claim 2 follows from the following Claim 3 and 4.

Claim 3. *[8] Suppose that the Decision (Modified) 3-party Diffie-Hellman, Bilinear Subgroup Decision, and Diffie-Hellman Subgroup Decision assumptions hold. Then for scheme Σ_{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 Claim 5.5, 5.6 and 5.7 in [8].

Claim 4. *Suppose that for scheme Σ_{AugBE} in [8] 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 Σ_{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 Σ_{A} with non-negligible advantage, we construct a PPT algorithm $\mathcal{B}$ to distinguish between H_2^{AugBE} and H_3^{AugBE} for Σ_{AugBE} with non-negligible advantage.

The game of $\mathcal{B}$ distinguishing between H_2^{AugBE} and H_3^{AugBE} is played in the subgroup $\mathbb{G}_{\tilde{N}}$ of order $\tilde{N} = qp$ in a composite group $\mathbb{G}_N$ of order $N = qpp_2p_3$. $\mathcal{B}$ is given the values of $\tilde{N}$, p_2 and p_3, but does not know the values of q or p. Since the game is played in the subgroup $\mathbb{G}_{\tilde{N}}$, $\mathcal{B}$ chooses for itself everything in the subgroup $\mathbb{G}_{p_3}$, i.e., $X_3 \in \mathbb{G}_{p_3}$.

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 $\{\mathsf{SK}_{(i,j)}^{\mathsf{AugBE}} | 1 \le i, j \le m\}$ as follows:

$$\begin{aligned}
\mathsf{PK}^{\mathsf{AugBE}} =& \big(\ g,\ h,\ G = g^{\delta} g_p^{\gamma},\ H = h^{\delta},\ G_q = g_q^{\beta}, \\
& \{G_i = g^{r_i}, G_{q,i} = g_q^{\beta r_i},\ H_i = h^{r_i}, H_{q,i} = h_q^{\beta r_i}, E_i = e(g,g)^{\alpha_i}, E_{q,i} = e(g_q, g_q)^{\beta \alpha_i}\}_{1 \le i \le m}, \\
& \{\bar{G}_j = g^{c_j},\ f_j,\ F_{q,j} = f_{q,j}^{\beta}\}_{1 \le j \le m}\ \big), \\
\mathsf{SK}_{(i,j)}^{\mathsf{AugBE}} =& \big(\tilde{K}_{i,j},\ \tilde{K}'_{i,j},\ \ \{\tilde{K}_{i,j,\tilde{j}}\}_{1 \le \tilde{j} \le m, \tilde{j} \ne j}\ \big) \\
=& \big(g^{\alpha_i} g^{r_i c_j} f_j^{\sigma_{i,j}},\ g^{\sigma_{i,j}},\ \ \{f_{\tilde{j}}^{\sigma_{i,j}}\}_{1 \le \tilde{j} \le m, \tilde{j} \ne j}\ \big),
\end{aligned}$$

where generators $g_q, h_q, f_{q,1}, \ldots, f_{q,m} \in \mathbb{G}_q$, $g_p, h_p, f_{p,1}, \ldots, f_{p,m} \in \mathbb{G}_p$ and exponents $\{r_i, \alpha_i \in \mathbb{Z}_{\tilde{N}}\}_{1 \le i \le m}$, $\{c_j \in \mathbb{Z}_{\tilde{N}}\}_{1 \le j \le m}$, $\beta \in \mathbb{Z}_q$, $\gamma \in \mathbb{Z}_p$, $\delta \in \mathbb{Z}_{\tilde{N}}$, $\sigma_{i,j} (1 \le i, j \le m) \in \mathbb{Z}_{\tilde{N}}$ are randomly chosen, and $g = g_q g_p, h = h_q h_p, f_j = f_{q,j} f_{p,j} (1 \le j \le m)$.

$\mathcal{B}$ randomly chooses $z_1, \ldots, z_m,\ u_x(x \in \mathbb{S}) \in \mathbb{Z}_N$, then gives $\mathcal{A}$ the following public parameters PK:

$$g,\ h,\ f = \prod_{1 \le j \le m} f_j,\ G,\ H,\ G_q,\ F_q = \prod_{1 \le j \le m} F_{q,j},$$
$$\{G_i,\ G_{q,i},\ H_i,\ H_{q,i},\ \hat{G}_i = g^{z_i},\ E_i,\ E_{q,i}\}_{1 \le i \le m},$$
$$\{\bar{G}_j\}_{1 \le j \le m},\ \ \{U_x = g^{u_x}\}_{x \in \mathbb{S}}.$$

Note that $\mathcal{B}$ implicitly chooses $f_q \in \mathbb{G}_q, f_p \in \mathbb{G}_p$ such that $f_q = \prod_{1 \le j \le m} f_{q,j},\ f_p = \prod_{1 \le j \le m} f_{p,j}$.

Key Query. $\mathcal{A}$ issues adaptive private key queries. To respond to a query for $((i,j), S_{(i,j)})$, $\mathcal{B}$ randomly chooses $R, R', R'', R_x(x \in S_{(i,j)}) \in \mathbb{G}_{p_3}$, and creates the private key $\mathsf{SK}_{(i,j),S_{(i,j)}}$ from $\mathsf{SK}^{\mathsf{AugBE}}_{(i,j)}$ as follows

$$\begin{aligned}\mathsf{SK}_{(i,j),S_{(i,j)}} &= \big(\ K_{i,j},\ K'_{i,j},\ K''_{i,j},\ \ \{K_{i,j,x}\}_{x \in S_{(i,j)}}\ \big) \\ &= \big(\ \tilde{K}_{i,j} \cdot \prod_{1 \le \tilde{j} \le m, \tilde{j} \ne j} \tilde{K}_{i,j,\tilde{j}} \cdot R,\ \tilde{K}'_{i,j} \cdot R',\ (\tilde{K}'_{i,j})^{z_i} \cdot R'',\ \ \{(\tilde{K}'_{i,j})^{u_x} \cdot R_x\}_{x \in S_{(i,j)}}\ \big).\end{aligned}$$

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 $CT^{\mathsf{AugBE}} = \langle (\tilde{R}_i, \tilde{\tilde{R}}_i, \tilde{Q}_i, \tilde{\tilde{Q}}_i, \tilde{T}_i)_{i=1}^m,\ (\tilde{C}_j, \tilde{\tilde{C}}_j)_{j=1}^m,\ Y \rangle$, which is encrypted to $(i^*, j^*) \in \{(\bar{i}, m+1), (\bar{i}+1, 1)\}$ and in the form of

1. For each $i \in \{1, \ldots, m\}$:
   - if $i > i^*$: $\tilde{R}_i = G_{q,i}^{s_i},\ \ \tilde{\tilde{R}}_i = H_{q,i}^{\kappa s_i},\ \ \tilde{Q}_i = G_q^{\tau s_i},\ \ \tilde{\tilde{Q}}_i = (\prod_{\hat{j} \in Y_i} F_{q,\hat{j}})^{\tau s_i},\ \ \tilde{T}_i = M \cdot E_{q,i}^{\tau s_i}$.
   - if $i = i^*$: $\tilde{R}_i = G_i^{s_i},\ \ \tilde{\tilde{R}}_i = H_i^{\kappa s_i},\ \ \tilde{Q}_i = g^{\tau s_i},\ \ \tilde{\tilde{Q}}_i = (\prod_{\hat{j} \in Y_i} f_{\hat{j}})^{\tau s_i},\ \ \tilde{T}_i = M \cdot E_i^{\tau s_i}$.
   - if $i < i^*$: $\tilde{R}_i = g^{\nu_{i,1}},\ \ \tilde{\tilde{R}}_i = h^{\kappa \nu_{i,1}},\ \ \tilde{Q}_i = g^{\nu_{i,2}},\ \ \tilde{\tilde{Q}}_i = (\prod_{\hat{j} \in Y_i} f_{\hat{j}})^{\nu_{i,2}},\ \ \tilde{T}_i = e(g,g)^{\nu_{i,3}}$.
2. For each $j \in \{1, \ldots, m\}$:
   - if $j \ge j^*$: $\tilde{C}_j = \bar{G}_j^{\tau} h^{\kappa t_j},\qquad \tilde{\tilde{C}}_j = g^{t_j}$.
   - if $j < j^*$: $\tilde{C}_j = \bar{G}_j^{\tau} h^{\kappa t_j} H^{\kappa \mu_j},\ \ \tilde{\tilde{C}}_j = g^{t_j} G^{\mu_j}$.

where $\kappa, \tau, s_{i^*}, \ldots, s_m, t_1, \ldots, t_m, \mu_1, \ldots, \mu_{j^*-1} \in \mathbb{Z}_{\tilde{N}}$, $(\nu_{1,1}, \nu_{1,2}, \nu_{1,3}), \ldots, (\nu_{i^*-1,1}, \nu_{i^*-1,2}, \nu_{i^*-1,3}) \in \mathbb{Z}_{\tilde{N}}^3$ are randomly chosen and Y_i denotes the set of all values j such that (i,j) in the set Y, i.e., $Y_i = \{j | (i,j) \in Y\}$.

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 $\tilde{\tilde{Q}}_i = (\prod_{\hat{j} \in Y_i} F_{q,\hat{j}})^{\tau s_i} = F_q^{\tau s_i}$ for $i > i^*$, $\tilde{\tilde{Q}}_i = (\prod_{\hat{j} \in Y_i} f_{\hat{j}})^{\tau s_i} = f^{\tau s_i}$ for $i = i^*$, and $\tilde{\tilde{Q}}_i = (\prod_{\hat{j} \in Y_i} f_{\hat{j}})^{\nu_{i,2}} = f^{\nu_{i,2}}$ for $i < i^*$. $\mathcal{B}$ constructs the LSSS matrix (A, ρ) for $\mathbb{A}_{S^*}$. Let $L \times D$ be the size of (A, ρ). $\mathcal{B}$ randomly chooses $d_1, \ldots, d_m \in \mathbb{Z}_N$, $(\xi_1, \ldots, \xi_L) \in \mathbb{Z}_N^L$, $\boldsymbol{v} = (\pi, v_2, \ldots, v_L) \in \mathbb{Z}_N^D$, then creates the ciphertext $\langle (A, \rho),\ (R_i, \bar{R}_i, Q_i, \bar{Q}_i, \hat{Q}_i, T_i)_{i=1}^m, (C_j, \bar{C}_j)_{j=1}^m, (P_l, \bar{P}_l)_{l=1}^L \rangle$ as follows:

1. For each $i \in \{1, \ldots, m\}$: $R_i = \tilde{R}_i,\ \bar{R}_i = \tilde{\tilde{R}}_i,\ Q_i = \tilde{Q}_i,\ \bar{Q}_i = \tilde{\tilde{Q}}_i \cdot \hat{G}_i^{d_i} f^{\pi},\ \hat{Q}_i = g^{d_i},\ T_i = \tilde{T}_i$.
2. For each $j \in \{1, \ldots, m\}$: $C_j = \tilde{C}_j,\ \bar{C}_j = \tilde{\tilde{C}}_j$.
3. For each $l \in \{1, \ldots, L\}$: $P_l = f^{A_l \cdot \boldsymbol{v}} U_{\rho(l)}^{-\xi_l},\ \ \bar{P}_l = g^{\xi_l}$.

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}.

Note that the distributions of the public parameters, private keys and challenge ciphertext that $\mathcal{B}$ gives $\mathcal{A}$ are same as the real scheme, $\mathcal{B}$'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}.