Traceable CP-ABE on Prime Order Groups: Fully Secure and Fully Collusion-Resistant Blackbox Traceable

Zhen Liu(✉) and Duncan S. Wong

Security and Data Sciences, ASTRI, Hong Kong SAR, China
zhenliu7-c@my.cityu.edu.hk, duncanwong@astri.org

Abstract. In Ciphertext-Policy Attribute-Based Encryption (CP-ABE), access policies associated with the ciphertexts are generally role-based and the attributes satisfying the policies are generally *shared* by multiple users. If a malicious user, with his attributes shared with multiple other users, created a decryption blackbox for sale, this malicious user could be difficult to identify from the blackbox. Hence in practice, a useful CP-ABE scheme should have some tracing mechanism to identify this 'traitor' from the blackbox. In this paper, we propose the first CP-ABE scheme which simultaneously achieves (1) fully collusion-resistant blackbox traceability in the standard model, (2) full security in the standard model, and (3) on prime order groups. When compared with the latest fully collusion-resistant blackbox traceable CP-ABE schemes, this new scheme achieves the same efficiency level, enjoying the sub-linear overhead of $O(\sqrt{N})$, where N is the number of users in the system. This new scheme is highly expressive and can take any monotonic access structures as ciphertext policies.

Keywords: Traceable · Ciphertext-policy Attribute Based Encryption · Prime order groups

1 Introduction

In a Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [1,7] system, each user possesses a set of attributes and a private key which is generated according to his attributes, and the encrypting party does not need to know or specify the exact identities of the targeted receivers, instead, the encrypting party can define an *access policy* over role-based/descriptive *attributes* to encrypt a message, so that only the users whose attributes satisfy the access policy can decrypt the ciphertext. For example, a school secretary, say Alice, may encrypt some messages using "(Mathematics AND (PhD Student OR Alumni))", which is an *access policy* defined over descriptive *attributes*, say "Mathematics", "PhD Student", and "Alumni", so that only PhD students and alumni in the Department of Mathematics have access to the messages. Due to the high flexibility and expressivity of the access policy, CP-ABE has promising applications related to access

S. Qing et al. (Eds.): ICICS 2015, LNCS 9543, pp. 109–124, 2016.
DOI: 10.1007/978-3-319-29814-6_10

control, such as secure cloud storage access and sharing, and has attracted great attention in the research community. Among the CP-ABE schemes recently proposed, [1,2,6,8,10,11,19–21], progress has been made on the schemes' security, access policy expressivity, and efficiency. While the schemes with practical security and expressivity (i.e. full security against adaptive adversaries in the standard model and high expressivity of supporting any monotone access structures) have been proposed in [10,11,19], the traceability of traitors which intentionally expose their decryption keys has been becoming an important concern related to the applicability of CP-ABE. Specifically, due to the nature of CP-ABE, access policies associated with the ciphertexts do not have to contain the exact identities of the eligible receivers. Instead, access policies are role-based and the attributes (and the corresponding decryption privilege) are generally *shared* by multiple users. As a result, a malicious user, with his attributes shared with multiple other users, might have an intention to leak the corresponding decryption key or some decryption privilege in the form of a decryption blackbox/device in which the decryption key is embedded, for example, for financial gain or for some other incentives, as there is little risk of getting caught. While all the aforementioned CP-ABE schemes lack the traitor tracing functionality, recently a handful of traceable CP-ABE schemes have been proposed in [3,13,14].

In the aforementioned non-traceable CP-ABE schemes, an easy and attractive way for a malicious user to make money is to sell a well-formed decryption key where the corresponding attribute set does not contain his identity-related attributes. For example, a malicious user with attributes {Bob, PhD, Mathematics} may build and sell a new decryption key with attributes {PhD, Mathematics}, and does not worry getting caught, since many other users share the attributes {PhD, Mathematics}. Liu et al. [14] proposed a whitebox traceable CP-ABE scheme that can deter users from such malicious behaviours, i.e., given a well-formed decryption key as input, a tracing algorithm can find out the malicious user who created the key from his/her original key. To avoid the whitebox traceability, instead of selling a well-formed decryption key, a more sophisticated malicious user may build and sell a decryption device/blackbox while keeping the embedded decryption key and algorithm hidden. Liu et al. [13] proposed a blackbox traceable CP-ABE scheme that can deter users from these more practical attacks, i.e., given a decryption blackbox/device, while the decryption key and even the decryption algorithm could be hidden, the tracing algorithm, which treats the decryption blackbox as an oracle, can still find out the malicious user whose key must have been used in constructing the decryption blackbox. Liu et al. proved that the CP-ABE scheme in [13] is fully secure in the standard model and fully collusion-resistant blackbox traceable in the standard model, where *fully collusion-resistant blackbox traceability* means that the number of colluding users in constructing a decryption blackbox is not limited and can be arbitrary. In addition, the scheme in [13] is highly expressive (i.e. supporting any monotonic access structures), and as a fully collusion-resistant blackbox traceable CP-ABE scheme, it achieves the most efficient level to date, i.e. the overhead for the fully collusion-resistant blackbox traceability is in $O(\sqrt{N})$,

where N is the number of users in the system. However, the scheme in [13] is based on composite order groups with order being the product of three large primes, and this severely limits its applicability. Liu and Wong [15] proposed a fully collusion-resistant blackbox traceable CP-ABE scheme on prime order groups, but achieves only selective security, where the adversary is required to declare his attacking target before seeing the system public key. Another recent blackbox traceable CP-ABE scheme is due to Deng et al. [3], which is only *t-collusion-resistant* traceable, where the number of colluding users is limited, i.e., less than a parameter t. In addition, the scheme in [3] is only selectively secure and the security is proven in the random oracle model.

1.1 Our Results

In this paper, we propose a new CP-ABE scheme that is fully secure in the standard model, fully collusion-resistant blackbox traceable in the standard model, and highly expressive (i.e. supporting any monotonic access structures). On the efficiency, as a fully collusion-resistant blackbox traceable CP-ABE scheme, this new scheme also achieves the most efficient level to date, i.e. the overhead for the fully collusion-resistant blackbox traceability is in $O(\sqrt{N})$. When compared with the CP-ABE scheme in [13], the advantage of this new scheme is that this scheme is constructed on prime order groups. Note that this implies this new scheme has better security and performance than the scheme in [13], although both of them are fully secure in the standard model and have overhead in $O(\sqrt{N})$. More specifically, as it has been shown (e.g. in [5,9]), the constructions on composite order groups will result in significant loss of efficiency and the security will rely on some non-standard assumptions (e.g. the Subgroup Decision Assumptions) and an additional assumption that the group order is hard to factor. To the best of our knowledge, this is the first CP-ABE scheme that is fully collusion-resistant blackbox traceable, fully secure, and constructed on prime order groups.

Related Work. In [13] Liu et al. defined a 'functional' CP-ABE that has the same functionality as the conventional CP-ABE (i.e. having all the appealing properties of the conventional CP-ABE), except that each user is assigned and identified by a unique index, which will enable the traceability of traitors. Liu et al. also defined the security and the fully collusion-resistant blackbox traceability for such a 'functional' CP-ABE. Furthermore, Liu et al. defined a new primitive called Augmented CP-ABE (AugCP-ABE) and formalized its security using message-hiding and index-hiding games. Then Liu et al. proved that *an AugCP-ABE scheme with message-hiding and index-hiding properties can be directly transferred to a secure CP-ABE with fully collusion-resistant blackbox traceability.* With such a framework, Liu et al. obtained a fully secure and fully collusion-resistant blackbox traceable CP-ABE scheme by constructing an AugCP-ABE scheme with message-hiding and index-hiding properties. It will be tempting to obtain a prime order construction by applying the existing general tools of converting constructions from composite order groups to prime order groups, e.g. [4,9], to the composite order group construction of [13]. However, as

the traceability is a new feature of CP-ABE and these tools focus on the conventional security (i.e. hiding the messages), it is not clear whether these tools are applicable to the traceable CP-ABE of [13].

Outline. In this paper, we also follow the framework in [13]. In particular, in Sect. 2 we review the definitions and security models of AugCP-ABE, then in Sect. 3 we propose our AugCP-ABE construction on prime order groups and prove that our AugCP-ABE construction is message-hiding and index-hiding in the standard model. As a result, we obtain a fully secure and fully collusion-resistant blackbox traceable CP-ABE scheme on prime order groups.

2 Augmented CP-ABE Definitions

In this section, we review the definitions of Augmented CP-ABE, which is proposed by Liu et al. [13] as a primitive that help constructing fully collusion-resistant blackbox traceable CP-ABE.

2.1 Definitions and Security Models

Given a positive integer n, let $[n]$ be the set $\{1, 2, \ldots, n\}$. An Augmented CP-ABE (AugCP-ABE) system consists of the following four algorithms:

$\mathsf{Setup}_{\mathsf{A}}(\lambda, \mathcal{U}, N) \rightarrow (\mathsf{PP}, \mathsf{MSK})$. The algorithm takes as input a security parameter λ, the attribute universe $\mathcal{U}$, and the number of users N in the system, then runs in polynomial time in λ, and outputs the public parameter PP and a master secret key MSK.

$\mathsf{KeyGen}_{\mathsf{A}}(\mathsf{PP}, \mathsf{MSK}, S) \rightarrow \mathsf{SK}_{k,S}$. The algorithm takes as input PP, MSK, and an attribute set S, and outputs a private key $\mathsf{SK}_{k,S}$, which is assigned and identified by a unique index $k \in [N]$.

$\mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, \mathbb{A}, \bar{k}) \rightarrow CT$. The algorithm takes as input PP, a message M, an access policy $\mathbb{A}$ over $\mathcal{U}$, and an index $\bar{k} \in [N+1]$, and outputs a ciphertext CT. $\mathbb{A}$ **is included in** CT, **but the value of** $\bar{k}$ **is not**.

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

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

Security. The security of AugCP-ABE is defined by the following three games, where the first two are for message-hiding, and the third one is for the index-hiding property. It is worth noticing that, as pointed in [13], in the three games: (1) the adversary is allowed to specify the index of the private key when it makes key queries for the attribute sets of its choice, i.e., for $t = 1$ to Q, the adversary submits (index, attribute set) pair (k_t, S_{k_t}) to query a private key for attribute set S_{k_t}, where $Q \leq N$, $k_t \in [N]$, and $k_t \neq k_{t'}$ $\forall 1 \leq t \neq t' \leq Q$ (this is to guarantee that each user/key can be *uniquely* identified by an index); and (2) for $k_t \neq k_{t'}$ we do not require $S_{k_t} \neq S_{k_{t'}}$, i.e., different users/keys may have the same attribute set.
In the following **message-hiding game** between a challenger and an adversary $\mathcal{A}$, $\bar{k} = 1$ (the first game, $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$) or $\bar{k} = N+1$ (the second game, $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_{N+1}}$).

Setup. The challenger runs $\mathsf{Setup}_{\mathsf{A}}(\lambda, \mathcal{U}, N)$ and gives the public parameter PP to $\mathcal{A}$.
Phase 1. For $t = 1$ to Q_1, $\mathcal{A}$ adaptively submits (index, attribute set) pair (k_t, S_{k_t}), and the challenger responds with a private key $\mathsf{SK}_{k_t, S_{k_t}}$.
Challenge. $\mathcal{A}$ submits two equal-length messages M_0, M_1 and an access policy $\mathbb{A}^*$. The challenger flips a random coin $b \in \{0, 1\}$, and sends $CT \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M_b, \mathbb{A}^*, \bar{k})$ to $\mathcal{A}$.
Phase 2. For $t = Q_1 + 1$ to Q, $\mathcal{A}$ adaptively submits (index, attribute set) pair (k_t, S_{k_t}), and the challenger responds with a private key $\mathsf{SK}_{k_t, S_{k_t}}$.
Guess. $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ for b.

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

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

Definition 1. *A N-user AugCP-ABE system is message-hiding if for all probabilistic polynomial time (PPT) adversaries $\mathcal{A}$ the advantages $\mathsf{MH}^{\mathsf{A}}_1\mathsf{Adv}_{\mathcal{A}}$ and $\mathsf{MH}^{\mathsf{A}}_{N+1}\mathsf{Adv}_{\mathcal{A}}$ are negligible in λ.*

$\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$. In the third game, **index-hiding game**, for any non-empty attribute set $S^* \subseteq \mathcal{U}$, we define **the strictest access policy** as $\mathbb{A}_{S^*} = \bigwedge_{x \in S^*} x$, and require that an adversary cannot distinguish between an encryption using $(\mathbb{A}_{S^*}, \bar{k})$ and $(\mathbb{A}_{S^*}, \bar{k} + 1)$ without a private decryption key $\mathsf{SK}_{\bar{k}, S_{\bar{k}}}$ such that $S_{\bar{k}} \supseteq S^*$. The game takes as input a parameter $\bar{k} \in [N]$ which is given to both the challenger and the adversary $\mathcal{A}$. The game proceeds as follows:

Setup. The challenger runs $\mathsf{Setup}_{\mathsf{A}}(\lambda, \mathcal{U}, N)$ and gives the public parameter PP to $\mathcal{A}$.

Key Query. For $t = 1$ to Q, $\mathcal{A}$ adaptively submits (index, attribute set) pair (k_t, S_{k_t}), and the challenger responds with a private key $\mathsf{SK}_{k_t,S_{k_t}}$.

Challenge. $\mathcal{A}$ submits a message M and a non-empty attribute set S^*. The challenger flips a random coin $b \in \{0,1\}$, and sends $CT \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, \mathbb{A}_{S^*}, \bar{k} + b)$ to $\mathcal{A}$.

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

$\mathcal{A}$ wins the game if $b' = b$ under the **restriction** that none of the queried pairs $\{(k_t, S_{k_t})\}_{1 \le t \le Q}$ can satisfy $(k_t = \bar{k}) \wedge (S_{k_t}$ satisfies $\mathbb{A}_{S^*})$, i.e. $(k_t = \bar{k}) \wedge (S_{k_t} \supseteq S^*)$. The advantage of $\mathcal{A}$ is defined as $\mathsf{IH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}] = |\Pr[b' = b] - \frac{1}{2}|$.

Definition 2. *A N-user AugCP-ABE system is index-hiding if for all PPT adversaries $\mathcal{A}$ the advantages $\mathsf{IH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}]$ for $\bar{k} = 1, \ldots, N$ are negligible in λ.*

2.2 The Reduction of Traceable CP-ABE to Augmented CP-ABE

Let $\Sigma_{\mathsf{A}} = (\mathsf{Setup}_{\mathsf{A}}, \mathsf{KeyGen}_{\mathsf{A}}, \mathsf{Encrypt}_{\mathsf{A}}, \mathsf{Decrypt}_{\mathsf{A}})$ be an Augmented CP-ABE, define $\mathsf{Encrypt}(\mathsf{PP}, M, \mathbb{A}) = \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, \mathbb{A}, 1)$, and let $\Sigma = (\mathsf{Setup}_{\mathsf{A}}, \mathsf{KeyGen}_{\mathsf{A}}, \mathsf{Encrypt}, \mathsf{Decrypt}_{\mathsf{A}})$. It is apparent that Σ is a 'functional' CP-ABE that has the same functionality as the conventional CP-ABE, except that the number of users in the system is predefined and each user is assigned a unique index. As shown in [13], with the Trace algorithm in [13, Sect. 3.2], Σ achieves fully collusion-resistant blackbox traceability against key-like decryption blackbox [1].

Theorem 1. *[13, Theorem 1] If Σ_{A} is message-hiding and index-hiding, then Σ is secure, and using the Trace algorithm, Σ is traceable against key-like decryption blackbox.*

3 An Augmented CP-ABE Construction on Prime Order Groups

Now we construct an AugCP-ABE scheme on prime order groups, and prove that this AugCP-ABE scheme is message-hiding and index-hiding in the standard model. Combined with the results in Sect. 2.2, we obtain a CP-ABE scheme that is fully collusion-resistant blackbox traceable in the standard model, fully secure in the standard model, and on prime order groups.

3.1 Preliminaries

Before proposing our AugCP-ABE scheme, we first review some preliminaries.

[1] Roughly speaking, a key-like decryption blackbox $\mathcal{D}$ is described by a non-empty attribute set $S_{\mathcal{D}}$ and a non-negligible probability value ϵ, and for any access policy $\mathbb{A}$, if it is satisfied by $S_{\mathcal{D}}$, this blackbox $\mathcal{D}$ can decrypt the ciphertexts associated with $\mathbb{A}$ with probability at least ϵ. Please refer to [13] for more formal details.

Bilinear Groups. Let $\mathcal{G}$ be a group generator, which takes a security parameter λ and outputs $(p, \mathbb{G}, \mathbb{G}_T, e)$ where p is a prime, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order p, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a map such that: (1) (Bilinear) $\forall g, h \in \mathbb{G}, a, b \in \mathbb{Z}_p, e(g^a, h^b) = e(g, h)^{ab}$, (2) (Non-Degenerate) $\exists g \in \mathbb{G}$ such that $e(g, g)$ has order p in $\mathbb{G}_T$. We refer to $\mathbb{G}$ as the *source group* and $\mathbb{G}_T$ as the *target group*. We assume that group operations in $\mathbb{G}$ and $\mathbb{G}_T$ as well as the bilinear map e are efficiently computable, and the description of $\mathbb{G}$ and $\mathbb{G}_T$ includes a generator of $\mathbb{G}$ and $\mathbb{G}_T$ respectively.

Complexity Assumptions. We will base the message-hiding property of our AugCP-ABE scheme on the Decisional Linear Assumption (DLIN), the Decisional 3-Party Diffie-Hellman Assumption (D3DH) and the Source Group q-Parallel BDHE Assumption, and will base the index-hiding property of our AugCP-ABE scheme on the DLIN assumption and the D3DH assumption. Note that the DLIN assumption and the D3DH assumption are standard and generally accepted assumptions, and the Source Group q-Parallel BDHE Assumption is introduced and proved by Lewko and Waters in [12]. Please refer to the full version [16, Appendix A] for the details of the three assumptions.

Dual Pairing Vector Spaces. Our construction will use dual pairing vector spaces, a tool introduced by Okamoto and Takashima [17–19] and developed by Lewko [9] and Lewko and Waters [12]. Let $\boldsymbol{v} = (v_1, \ldots, v_n)$ be a vector over $\mathbb{Z}_p$, the notation $g^{\boldsymbol{v}}$ denotes a tuple of group elements as $g^{\boldsymbol{v}} := (g^{v_1}, \ldots, g^{v_n})$. Furthermore, for any $a \in \mathbb{Z}_p$ and $\boldsymbol{v} = (v_1, \ldots, v_n), \boldsymbol{w} = (w_1, \ldots, w_n) \in \mathbb{Z}_p^n$, define $(g^{\boldsymbol{v}})^a := g^{a\boldsymbol{v}} = (g^{av_1}, \ldots, g^{av_n})$, $g^{\boldsymbol{v}} g^{\boldsymbol{w}} := g^{\boldsymbol{v}+\boldsymbol{w}} = (g^{v_1+w_1}, \ldots, g^{v_n+w_n})$, and define a bilinear map e_n on n-tuples of $\mathbb{G}$ as $e_n(g^{\boldsymbol{v}}, g^{\boldsymbol{w}}) := \prod_{i=1}^n e(g^{v_i}, g^{w_i}) = e(g, g)^{(\boldsymbol{v} \cdot \boldsymbol{w})}$, where the dot/inner product $\boldsymbol{v} \cdot \boldsymbol{w}$ is computed modulo p.

For a fixed (constant) dimension n, we say two bases $\mathbb{B} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ and $\mathbb{B}^* := (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*)$ of $\mathbb{Z}_p^n$ are "dual orthonormal" when $\boldsymbol{b}_i \cdot \boldsymbol{b}_j^* \equiv 0 (\text{mod } p)$ $\forall 1 \leq i \neq j \leq n$ and $\boldsymbol{b}_i \cdot \boldsymbol{b}_i^* \equiv \psi (\text{mod } p)$ $\forall 1 \leq i \leq n$, where ψ is a non-zero element of $\mathbb{Z}_p$. (This is a slight abuse of the terminology "orthonormal", since ψ is not constrained to be 1.) For a generator $g \in \mathbb{G}$, we note that $e_n(g^{\boldsymbol{b}_i}, g^{\boldsymbol{b}_j^*}) = 1$ whenever $i \neq j$, where 1 here denotes the identity element in $\mathbb{G}_T$. Let $Dual(\mathbb{Z}_p^n, \psi)$ denote the set of pairs of dual orthonormal bases of dimension n with dot products $\boldsymbol{b}_i \cdot \boldsymbol{b}_i^* = \psi$, and $(\mathbb{B}, \mathbb{B}^*) \xleftarrow{R} Dual(\mathbb{Z}_p^n, \psi)$ denote choosing a random pair of bases from this set. As our AugCP-ABE construction will use dual pairing vector spaces, the security proof will use a lemma and a Subspace Assumption, which are introduced and proved by Lewko and Waters [12], in the setting of dual pairing vector spaces. Please refer to the full version [16, Appendix A.1] for the details of this lemma and the Subspace Assumption. Here we would like to stress that *the Subspace Assumption is implied by DLIN assumption.*

To construct our AugCP-ABE scheme, we further define a new notation. In particular, for any $\boldsymbol{v} = (v_1, \ldots, v_n) \in \mathbb{Z}_p^n$, $\boldsymbol{v'} = (v'_1, \ldots, v'_{n'}) \in \mathbb{Z}_p^{n'}$, we define

$$(g^{\boldsymbol{v}})^{\boldsymbol{v'}} := ((g^{\boldsymbol{v}})^{v'_1}, \ldots, (g^{\boldsymbol{v}})^{v'_{n'}}) = (g^{v'_1 v_1}, \ldots, g^{v'_1 v_n}, \ldots, g^{v'_{n'} v_1}, \ldots, g^{v'_{n'} v_n}) \in \mathbb{G}^{nn'}.$$

Note that for any $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{Z}_p^n, \boldsymbol{v'}, \boldsymbol{w'} \in \mathbb{Z}_p^{n'}$, we have

$$e_{nn'}((g^{\boldsymbol{v}})^{\boldsymbol{v'}}, (g^{\boldsymbol{w}})^{\boldsymbol{w'}}) = \prod_{j=1}^{n'} \prod_{i=1}^{n} e(g^{v'_j v_i}, g^{w'_j w_i}) = e(g,g)^{(\boldsymbol{v}\cdot\boldsymbol{w})(\boldsymbol{v'}\cdot\boldsymbol{w'})}.$$

Linear Secret-Sharing Schemes (LSSS). As of previous work, we use linear secret-sharing schemes (LSSS) to express the access policies. An LSSS is a share-generating matrix A whose rows are labeled by attributes via a function ρ. An attribute set S satisfies the LSSS access matrix (A, ρ) if the rows labeled by the attributes in S have the *linear reconstruction* property, namely, there exist constants $\{\omega_i | \rho(i) \in S\}$ such that, for any valid shares $\{\lambda_i\}$ of a secret s, we have $\sum_{\rho(i)\in S} \omega_i \lambda_i = s$. Please refer to the full version [16, Appendix D] for the formal definitions of access structures and LSSS.

Notations. Suppose the number of users N in the system equals n^2 for some n[2]. We arrange the users in a $n \times n$ matrix and uniquely assign a tuple (i, j) where $1 \le i, j \le n$, to each user. A user at position (i, j) of the matrix has index $k = (i-1) * n + j$. For simplicity, we directly use (i, j) as the index where $(i, j) \ge (\bar{i}, \bar{j})$ means that $((i > \bar{i}) \vee (i = \bar{i} \wedge j \ge \bar{j}))$. The use of pairwise notation (i, j) is purely a notational convenience, as $k = (i-1) * n + j$ defines a bijection between $\{(i, j) | 1 \le i, j \le n\}$ and $\{1, \ldots, N\}$. We conflate the notation and consider the attribute universe to be $[\mathcal{U}] = \{1, 2 \ldots, \mathcal{U}\}$, so $\mathcal{U}$ servers both as a description of the attribute universe and as a count of the total number of attributes. Given a bilinear group order p, one can randomly choose $r_x, r_y, r_z \in \mathbb{Z}_p$, and set $\boldsymbol{\chi}_1 = (r_x, 0, r_z)$, $\boldsymbol{\chi}_2 = (0, r_y, r_z)$, $\boldsymbol{\chi}_3 = \boldsymbol{\chi}_1 \times \boldsymbol{\chi}_2 = (-r_y r_z, -r_x r_z, r_x r_y)$. Let $span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$ be the subspace spanned by $\boldsymbol{\chi}_1$ and $\boldsymbol{\chi}_2$, i.e. $span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\} = \{\nu_1 \boldsymbol{\chi}_1 + \nu_2 \boldsymbol{\chi}_2 | \nu_1, \nu_2 \in \mathbb{Z}_p\}$. We can see that $\boldsymbol{\chi}_3$ is orthogonal to the subspace $span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$ and that $\mathbb{Z}_p^3 = span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2, \boldsymbol{\chi}_3\} = \{\nu_1 \boldsymbol{\chi}_1 + \nu_2 \boldsymbol{\chi}_2 + \nu_3 \boldsymbol{\chi}_3 | \nu_1, \nu_2, \nu_3 \in \mathbb{Z}_p\}$. For any $\boldsymbol{v} \in span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$, we have $(\boldsymbol{\chi}_3 \cdot \boldsymbol{v}) = 0$, and for random $\boldsymbol{v} \in \mathbb{Z}_p^3$, $(\boldsymbol{\chi}_3 \cdot \boldsymbol{v}) \neq 0$ happens with overwhelming probability.

3.2 AugCP-ABE Construction

$\mathsf{Setup}_{\mathsf{A}}(\lambda, \mathcal{U}, N = n^2) \to (\mathsf{PP}, \mathsf{MSK})$. The algorithm chooses a bilinear group $\mathbb{G}$ of order p and two generators $g, h \in \mathbb{G}$. It randomly chooses $(\mathbb{B}, \mathbb{B}^*), (\mathbb{B}_0, \mathbb{B}_0^*) \in Dual(\mathbb{Z}_p^3, \psi)$ and $(\mathbb{B}_1, \mathbb{B}_1^*), \ldots, (\mathbb{B}_{\mathcal{U}}, \mathbb{B}_{\mathcal{U}}^*) \in Dual(\mathbb{Z}_p^6, \psi)$. We let $\boldsymbol{b}_j, \boldsymbol{b}_j^* (1 \le j \le 3)$ denote the basis vectors belonging to $(\mathbb{B}, \mathbb{B}^*)$, $\boldsymbol{b}_{0,j}, \boldsymbol{b}_{0,j}^* (1 \le j \le 3)$ denote the basis vectors belonging to $(\mathbb{B}_0, \mathbb{B}_0^*)$, and $\boldsymbol{b}_{x,j}, \boldsymbol{b}_{x,j}^* (1 \le j \le 6)$ denote the basis vectors belonging to $(\mathbb{B}_x, \mathbb{B}_x^*)$ for each $x \in [\mathcal{U}]$. The algorithm also chooses random exponents $\alpha_1, \alpha_2 \in \mathbb{Z}_p$, $\{r_i, z_i, \alpha_{i,1}, \alpha_{i,2} \in$

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

$\mathbb{Z}_p\}_{i\in[n]},\ \{c_{j,1}, c_{j,2},\ y_j \in \mathbb{Z}_p\}_{j\in[n]}$. The public parameter PP is set to

$$\begin{aligned}\mathsf{PP} = \Big(& (p, \mathbb{G}, \mathbb{G}_T, e),\ g, h,\ g^{\boldsymbol{b}_1}, g^{\boldsymbol{b}_2}, h^{\boldsymbol{b}_1}, h^{\boldsymbol{b}_2}, h^{\boldsymbol{b}_{0,1}}, h^{\boldsymbol{b}_{0,2}},\\ & \{h^{\boldsymbol{b}_{x,1}}, h^{\boldsymbol{b}_{x,2}}, h^{\boldsymbol{b}_{x,3}}, h^{\boldsymbol{b}_{x,4}}\}_{x\in[\mathcal{U}]},\ \ F_1 = e(g,h)^{\psi\alpha_1},\ F_2 = e(g,h)^{\psi\alpha_2},\\ & \{E_{i,1} = e(g,g)^{\psi\alpha_{i,1}},\ E_{i,2} = e(g,g)^{\psi\alpha_{i,2}}\}_{i\in[n]},\\ & \{\boldsymbol{G}_i = g^{r_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},\ \boldsymbol{Z}_i = g^{z_i(\boldsymbol{b}_1+\boldsymbol{b}_2)}\}_{i\in[n]},\\ & \{\boldsymbol{H}_j = g^{c_{j,1}\boldsymbol{b}_1^* + c_{j,2}\boldsymbol{b}_2^*},\ \ \boldsymbol{Y}_j = \boldsymbol{H}_j^{y_j}\}_{j\in[n]}\Big).\end{aligned}$$

The master secret key is set to

$$\begin{aligned}\mathsf{MSK} = \Big(& \boldsymbol{b}_1^*, \boldsymbol{b}_2^*,\ \boldsymbol{b}_{0,1}^*, \boldsymbol{b}_{0,2}^*,\ \{\boldsymbol{b}_{x,1}^*, \boldsymbol{b}_{x,2}^*, \boldsymbol{b}_{x,3}^*, \boldsymbol{b}_{x,4}^*\}_{x\in[\mathcal{U}]},\\ & \alpha_1, \alpha_2,\ \{r_i, z_i,\ \alpha_{i,1}, \alpha_{i,2}\}_{i\in[n]},\ \{c_{j,1}, c_{j,2}\}_{j\in[n]}\Big).\end{aligned}$$

In addition, a counter $ctr = 0$ is implicitly included in MSK.

$\mathsf{KeyGen}_\mathsf{A}(\mathsf{PP}, \mathsf{MSK}, S) \to \mathsf{SK}_{(i,j),S}$. The algorithm first sets $ctr = ctr + 1$ and computes the corresponding index in the form of (i, j) where $1 \le i, j \le n$ and $(i-1) * n + j = ctr$. Then it randomly chooses $\sigma_{i,j,1}, \sigma_{i,j,2}, \delta_{i,j,1}, \delta_{i,j,2} \in \mathbb{Z}_p$, and outputs a private key $\mathsf{SK}_{(i,j),S} =$

$$\begin{aligned}\langle\ & (i,j), S,\ \boldsymbol{K}_{i,j} = g^{(\alpha_{i,1}+r_ic_{j,1})\boldsymbol{b}_1^* + (\alpha_{i,2}+r_ic_{j,2})\boldsymbol{b}_2^*} h^{(\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{b}_1^* + (\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{b}_2^*},\\ & \boldsymbol{K}'_{i,j} = g^{(\alpha_1+\sigma_{i,j,1}+\delta_{i,j,1})\boldsymbol{b}_1^* + (\alpha_2+\sigma_{i,j,2}+\delta_{i,j,2})\boldsymbol{b}_2^*},\ \boldsymbol{K}''_{i,j} = (\boldsymbol{K}'_{i,j})^{z_i},\\ & \boldsymbol{K}_{i,j,0} = g^{\delta_{i,j,1}\boldsymbol{b}_{0,1}^* + \delta_{i,j,2}\boldsymbol{b}_{0,2}^*},\ \{\boldsymbol{K}_{i,j,x} = g^{\sigma_{i,j,1}(\boldsymbol{b}_{x,1}^* + \boldsymbol{b}_{x,2}^*) + \sigma_{i,j,2}(\boldsymbol{b}_{x,3}^* + \boldsymbol{b}_{x,4}^*)}\}_{x\in S}\ \rangle.\end{aligned}$$

$\mathsf{Encrypt}_\mathsf{A}(\mathsf{PP}, M, \mathbb{A} = (A, \rho), (\bar{i}, \bar{j})) \to CT$. A is an $l \times m$ LSSS matrix and ρ maps each row A_k of A to an attribute $\rho(k) \in [\mathcal{U}]$. The algorithm first chooses random $\kappa,\ \tau,\ s_1, \ldots, s_n,\ t_1, \ldots, t_n \in \mathbb{Z}_p,\ \boldsymbol{v}_c,\ \boldsymbol{w}_1, \ldots, \boldsymbol{w}_n \in \mathbb{Z}_p^3,\ \xi_{1,1}, \xi_{1,2}, \ldots, \xi_{l,1},\ \xi_{l,2} \in \mathbb{Z}_p,\ \boldsymbol{u}_1, \boldsymbol{u}_2 \in \mathbb{Z}_p^m$. It also chooses random $r_x, r_y, r_z \in \mathbb{Z}_p$, and sets $\boldsymbol{\chi}_1 = (r_x, 0, r_z), \boldsymbol{\chi}_2 = (0, r_y, r_z), \boldsymbol{\chi}_3 = (-r_yr_z, -r_xr_z, r_xr_y)$. Then it randomly chooses $\boldsymbol{v}_i \in \mathbb{Z}_p^3$ *for* $i = 1, \ldots, \bar{i},\ \boldsymbol{v}_i \in span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$ *for* $i = \bar{i}+1, \ldots, n$. Let π_1 and π_2 be the first entries of $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ respectively. The algorithm creates a ciphertext $\langle (A, \rho),\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, \boldsymbol{Q}_i, \boldsymbol{Q}'_i, \boldsymbol{Q}''_i, T_i)_{i=1}^n,\ (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^n,\ (\boldsymbol{P}_k)_{k=0}^l \rangle$:

1. For each row $i \in [n]$:
   - if $i < \bar{i}$: choose random $\hat{s}_i \in \mathbb{Z}_p$, then set

$$\begin{aligned}& \boldsymbol{R}_i = (g^{\boldsymbol{b}_1+\boldsymbol{b}_2})^{\boldsymbol{v}_i},\ \ \boldsymbol{R}'_i = \boldsymbol{R}_i^{\kappa}, \qquad \boldsymbol{Q}_i = g^{s_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},\\ & \boldsymbol{Q}'_i = h^{s_i(\boldsymbol{b}_1+\boldsymbol{b}_2)} \boldsymbol{Z}_i^{t_i} h^{\pi_1\boldsymbol{b}_1+\pi_2\boldsymbol{b}_2}, \qquad \boldsymbol{Q}''_i = g^{t_i(\boldsymbol{b}_1+\boldsymbol{b}_2)}, \quad T_i = e(g,g)^{\hat{s}_i}.\end{aligned}$$

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

$$\begin{aligned}& \boldsymbol{R}_i = (\boldsymbol{G}_i)^{s_i\boldsymbol{v}_i},\ \ \boldsymbol{R}'_i = \boldsymbol{R}_i^{\kappa}, \qquad \boldsymbol{Q}_i = g^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)(\boldsymbol{b}_1+\boldsymbol{b}_2)},\\ & \boldsymbol{Q}'_i = h^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)(\boldsymbol{b}_1+\boldsymbol{b}_2)} \boldsymbol{Z}_i^{t_i} h^{\pi_1\boldsymbol{b}_1+\pi_2\boldsymbol{b}_2},\ \ \boldsymbol{Q}''_i = g^{t_i(\boldsymbol{b}_1+\boldsymbol{b}_2)},\\ & T_i = M \frac{(E_{i,1}E_{i,2})^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)}}{(F_1F_2)^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)} F_1^{\pi_1} F_2^{\pi_2}}.\end{aligned}$$

2. For each column $j \in [n]$:
   - if $j < \bar{j}$: choose random $\mu_j \in \mathbb{Z}_p$, then set $\boldsymbol{C}_j = (\boldsymbol{H}_j)^{\tau(\boldsymbol{v}_c+\mu_j\boldsymbol{\chi}_3)}(\boldsymbol{Y}_j)^{\kappa\boldsymbol{w}_j}$, $\boldsymbol{C}'_j = (\boldsymbol{Y}_j)^{\boldsymbol{w}_j}$.
   - if $j \geq \bar{j}$: set $\boldsymbol{C}_j = (\boldsymbol{H}_j)^{\tau\boldsymbol{v}_c}(\boldsymbol{Y}_j)^{\kappa\boldsymbol{w}_j}, \boldsymbol{C}'_j = (\boldsymbol{Y}_j)^{\boldsymbol{w}_j}$.
3. $\boldsymbol{P}_0 = h^{\pi_1\boldsymbol{b}_{0,1}+\pi_2\boldsymbol{b}_{0,2}}$, $\{\boldsymbol{P}_k = h^{(A_k\cdot\boldsymbol{u}_1+\xi_{k,1})\boldsymbol{b}_{\rho(k),1}-\xi_{k,1}\boldsymbol{b}_{\rho(k),2}+(A_k\cdot\boldsymbol{u}_2+\xi_{k,2})\boldsymbol{b}_{\rho(k),3}-\xi_{k,2}\boldsymbol{b}_{\rho(k),4}}\}_{k\in[l]}$.

$\mathsf{Decrypt}_{\mathsf{A}}(\mathsf{PP}, CT, \mathsf{SK}_{(i,j),S}) \to M$ or $\perp$. If the private key's attribute set S does not satisfy the ciphertext's LSSS (A, ρ), the algorithm outputs $\perp$, otherwise

1. Compute constants $\{\omega_k \in \mathbb{Z}_p | \rho(k) \in S\}$ such that $\sum_{\rho(k)\in S}\omega_k A_k = (1, 0, \ldots, 0)$, then compute $D_P = e_3(\boldsymbol{K}_{i,j,0}, \boldsymbol{P}_0) \prod_{\rho(k)\in S} e_6(\boldsymbol{K}_{i,j,\rho(k)}, \boldsymbol{P}_k)^{\omega_k}$.
2. Compute $D_I = \frac{e_3(\boldsymbol{K}_{i,j},\boldsymbol{Q}_i)\cdot e_3(\boldsymbol{K}''_{i,j},\boldsymbol{Q}''_i)\cdot e_9(\boldsymbol{R}'_i,\boldsymbol{C}'_j)}{e_3(\boldsymbol{K}'_{i,j},\boldsymbol{Q}'_i)\cdot e_9(\boldsymbol{R}_i,\boldsymbol{C}_j)}$.
3. Compute $M = T_i/(D_P \cdot D_I)$ as the output message. Assume the ciphertext is generated from message M' and index $(\bar{i}, \bar{j})$, it can be verified that only when $(i > \bar{i})$ or $(i = \bar{i} \wedge j \geq \bar{j})$, $M = M'$ will hold. This follows from the facts that for $i > \bar{i}$, we have $(\boldsymbol{v}_i \cdot \boldsymbol{\chi}_3) = 0$ (since $\boldsymbol{v}_i \in span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$), and for $i = \bar{i}$, we have that $(\boldsymbol{v}_i \cdot \boldsymbol{\chi}_3) \neq 0$ happens with overwhelming probability (since $\boldsymbol{v}_i$ is randomly chosen from $\mathbb{Z}_p^3$). The correctness details can be found in the full version [16, Appendix B].

Remarks: We borrow the ideas of [12, Sect. 5] to achieve the full security for prime order group constructions, and borrow the ideas of [13] to achieve fully collusion-resistant blackbox traceability. But the above construction and the later security proof are not trivial combinations of the two schemes. In particular, the public parameter components $g^{\boldsymbol{b}_1}, g^{\boldsymbol{b}_2}, h^{\boldsymbol{b}_1}, h^{\boldsymbol{b}_2}, h^{\boldsymbol{b}_{0,1}}, h^{\boldsymbol{b}_{0,2}}$, $\{h^{\boldsymbol{b}_{x,1}}, h^{\boldsymbol{b}_{x,2}}, h^{\boldsymbol{b}_{x,3}}, h^{\boldsymbol{b}_{x,4}}\}_{x\in[\mathcal{U}]}, F_1, F_2$, the key components $\boldsymbol{K}'_{i,j}, \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x\in S}$, and ciphertext components $\boldsymbol{P}_0, \{\boldsymbol{P}_k\}_{k\in[l]}$ are designed using the ideas of [12, Sect. 5]. To achieve fully collusion-resistant blackbox traceability, $\{E_{i,1}, E_{i,2}, \boldsymbol{G}_i, \boldsymbol{Z}_i\}_{i\in[n]}$, $\{\boldsymbol{H}_j\}_{j\in[n]}$ are put in the public parameter, and $\boldsymbol{K}_{i,j}, \boldsymbol{K}''_{i,j}$ are introduced into the private key. Note that $\boldsymbol{G}_i$ and $\boldsymbol{H}_j$ will be used to generate ciphertext components $\boldsymbol{R}_i$ and $\boldsymbol{C}_j$ respectively, and $e_9(\boldsymbol{R}_i, \boldsymbol{C}_j)$ will be computed during decryption, so that $\boldsymbol{G}_i$ and $\boldsymbol{H}_j$ must use the basis vectors of a pair of dual orthonormal bases, i.e. $\boldsymbol{G}_i$ uses $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ and $\boldsymbol{H}_j$ uses $(\boldsymbol{b}^*_1, \boldsymbol{b}^*_2)$. This prevents us from trivially using the proof of [12, Sect. 5], because in the construction of [12, Sect. 5], only $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_{0,1}, \boldsymbol{b}_{0,2}, \{\boldsymbol{b}_{x,1}, \boldsymbol{b}_{x,2}, \boldsymbol{b}_{x,3}, \boldsymbol{b}_{x,4}\}_{x\in[\mathcal{U}]}$ appear in the exponents of the public parameter components. As an informal evidence, while the AugCP-ABE scheme of [13] reduces its message-hiding property (in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$) to the security of the CP-ABE scheme of [11], it is impossible to make a similar reduction here, since the public parameter of the above AugCP-ABE construction contains $(\boldsymbol{b}^*_1, \boldsymbol{b}^*_2)$ while the public parameter of [12, Sect. 5] does not contain them. To address this problem, we introduce a new and crucial public parameter component $\boldsymbol{Y}_j = \boldsymbol{H}_j^{y_j}$ which does not have counterpart in the AugCP-ABE scheme of [13] or the CP-ABE scheme in [12, Sect. 5], and we reduce the message-hiding property of our construction directly to the underlying assumptions.

3.3 Security of the AugCP-ABE Construction

The following Theorems 2 and 3 show that our AugCP-ABE construction is message-hiding, and Theorem 4 shows that our AugCP-ABE construction is index-hiding.

Theorem 2. *Suppose the DLIN assumption, the D3DH assumption, and the source group q-parallel BDHE assumption hold. Then no PPT adversary can win* $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$ *with non-negligible advantage.*

Proof. Our message-hiding proof route here is quite similar to the security proof route of the conventional CP-ABE scheme by Lewko and Waters [12, Sect. 5]. But as discussed previously, this is not a trivial work.

We begin by defining our various types of semi-functional keys and ciphertexts. The semi-functional space in the exponent will correspond to the span of $\boldsymbol{b}_3, \boldsymbol{b}^*_3$, the span of $\boldsymbol{b}_{0,3}, \boldsymbol{b}^*_{0,3}$ and the span of each $\boldsymbol{b}_{x,5}, \boldsymbol{b}_{x,6}, \boldsymbol{b}^*_{x,5}, \boldsymbol{b}^*_{x,6}$.

Semi-functional Keys. To produce a semi-functional key for an attribute set S, one first calls the normal key generation algorithm to produce a normal key consisting of $\boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j}, \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x\in S}$ with index (i,j). One then chooses random value γ. The semi-functional key is

$$\boldsymbol{K}_{i,j}h^{\gamma\boldsymbol{b}^*_3},\ \boldsymbol{K}'_{i,j}g^{\gamma\boldsymbol{b}^*_3},\ \boldsymbol{K}''_{i,j}g^{z_i\gamma\boldsymbol{b}^*_3},\ \boldsymbol{K}_{i,j,0},\ \{\boldsymbol{K}_{i,j,x}\}_{x\in S}.$$

Semi-functional Ciphertexts. To produce a semi-functional ciphertext for an LSSS matrix (A,ρ) of size $l\times m$, one first calls the normal encryption algorithm to produce a normal ciphertext consisting of $\langle (A,\rho),\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, \boldsymbol{Q}_i, \boldsymbol{Q}'_i, \boldsymbol{Q}''_i, T_i)^n_{i=1}, (\boldsymbol{C}_j, \boldsymbol{C}'_j)^n_{j=1}, (\boldsymbol{P}_k)^l_{k=0}\rangle$. One then chooses random values $\pi_3, \xi_{k,3}(1\le k\le l)\in\mathbb{Z}_p$ and a random vector $\boldsymbol{u}_3\in\mathbb{Z}^m_p$ with first entry equal to π_3. The semi-functional ciphertext is:

$$\begin{aligned}&\langle (A,\rho),\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, \boldsymbol{Q}_i, \boldsymbol{Q}'_i h^{\pi_3\boldsymbol{b}_3}, \boldsymbol{Q}''_i, T_i)^n_{i=1}, (\boldsymbol{C}_j, \boldsymbol{C}'_j)^n_{j=1},\\ &\boldsymbol{P}_0 h^{\pi_3\boldsymbol{b}_{0,3}},\ (\boldsymbol{P}_k h^{(A_k\cdot\boldsymbol{u}_3+\xi_{k,3})\boldsymbol{b}_{\rho(k),5}-\xi_{k,3}\boldsymbol{b}_{\rho(k),6}})^l_{k=1}\rangle.\end{aligned}$$

Our proof is obtained via a hybrid argument over a sequence of games:

Game_{real}: The real message-hiding game $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_1}$ as defined in the Sect. 2.1.

Game_t $(0\le t\le Q)$: Let Q denote the total number of key queries that the attacker makes. For each t from 0 to Q, we define Game_t as follows: In Game_t, the ciphertext given to the attacker is semi-functional, as are the first t keys. The remaining keys are normal.

Game_{final}: In this game, all of the keys given to the attacker are semi-functional, and the ciphertext given to the attacker is a semi-functional encryption of a *random message*.

The outer structure of our hybrid argument will progress as shown in Fig. 1. First, we transition from Game_{real} to Game_0, then to Game_1, next to Game_2, and so on. We ultimately arrive at Game_Q, where the ciphertext and all of the keys given to the attacker are semi-functional. We then transition to Game_{final}, which is defined to be like Game_Q, except that the ciphertext given to the attacker is a semi-functional encryption of a random message. This will complete our proof, since any attacker has a zero advantage in this final game.

The transitions from Game_{real} to Game_0 and from Game_Q to Game_{final} are relatively easy, and can be accomplished directly via computational assumptions. The transitions from Game_{t-1} to Game_t require more intricate arguments. For these steps, we will need to treat Phase 1 key requests (before the challenge ciphertext) and Phase 2 key requests (after the challenge ciphertext) differently. We will also need to define two additional types of semi-functional keys:

Nominal Semi-functional Keys. To produce a nominal semi-functional key for an attribute set S, one first calls the normal key generation algorithm to produce a normal key consisting of $\boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j}, \boldsymbol{K}_{i,j,0}, \{\boldsymbol{K}_{i,j,x}\}_{x\in S}$ with index (i,j). One then chooses random values $\sigma_{i,j,3}, \delta_{i,j,3} \in \mathbb{Z}_p$. The nominal semi-functional key is:

$$\boldsymbol{K}_{i,j} h^{(\sigma_{i,j,3}+\delta_{i,j,3})\boldsymbol{b}^*_3},\ \boldsymbol{K}'_{i,j} g^{(\sigma_{i,j,3}+\delta_{i,j,3})\boldsymbol{b}^*_3},\ \boldsymbol{K}''_{i,j} g^{z_i(\sigma_{i,j,3}+\delta_{i,j,3})\boldsymbol{b}^*_3},$$
$$\boldsymbol{K}_{i,j,0} g^{\delta_{i,j,3}\boldsymbol{b}^*_{0,3}},\quad \{\boldsymbol{K}_{i,j,x} g^{\sigma_{i,j,3}(\boldsymbol{b}^*_{x,5}+\boldsymbol{b}^*_{x,6})}\}_{x\in S}.$$

We note that a nominal semi-functional key still correctly decrypts a semi-functional ciphertext.

Temporary Semi-functional Keys. A temporary semi-functional key is similar to a nominal semi-functional key, except that the semi-functional component attached to $\boldsymbol{K}'_{i,j}$ will now be randomized (this will prevent correct decryption of a semi-functional ciphertext) and $\boldsymbol{K}_{i,j}$ and $\boldsymbol{K}''_{i,j}$ change accordingly. More formally, to produce a temporary semi-functional key for an attribute set S, one first calls the normal key generation algorithm to produce a normal key consisting of $\boldsymbol{K}_{i,j}, \boldsymbol{K}'_{i,j}, \boldsymbol{K}''_{i,j}, \boldsymbol{K}_{i,j,0},\ \{\boldsymbol{K}_{i,j,x}\}_{x\in S}$ with index (i,j). One then chooses random values $\sigma_{i,j,3}, \delta_{i,j,3}, \gamma \in \mathbb{Z}_p$. The temporary semi-functional key is formed as:

$$\boldsymbol{K}_{i,j} h^{\gamma\boldsymbol{b}^*_3},\ \boldsymbol{K}'_{i,j} g^{\gamma\boldsymbol{b}^*_3},\ \boldsymbol{K}''_{i,j} g^{z_i\gamma\boldsymbol{b}^*_3},\ \boldsymbol{K}_{i,j,0} g^{\delta_{i,j,3}\boldsymbol{b}^*_{0,3}},\ \{\boldsymbol{K}_{i,j,x} g^{\sigma_{i,j,3}(\boldsymbol{b}^*_{x,5}+\boldsymbol{b}^*_{x,6})}\}_{x\in S}.$$

For each t from 1 to Q, we define the following additional games:

Game^N_t: This is like Game_t, except that the t^{th} key given to the attacker is a nominal semi-functional key. The first $t-1$ keys are still semi-functional in the original sense, while the remaining keys are normal.

Game^T_t: This is like Game_t, except that the t^{th} key given to the attacker is a temporary semi-functional key. The first $t-1$ keys are still semi-functional in the original sense, while the remaining keys are normal.

In order to transition from Game_{t-1} to Game_t in our hybrid argument, we will transition first from Game_{t-1} to Game^N_t, then to Game^T_t, and finally to

Game_t. The transition from Game_t^N to Game_t^T will require different computational assumptions for Phase 1 and Phase 2 queries (As shown in Fig. 1, we use two lemmas based on different assumptions to obtain the transition).

As shown in Fig. 1, we use a series of lemmas, i.e. Lemmas 4, 5, 6, 7, 8, and 9, to prove the transitions. The details of these lemmas and their proofs can be found in the full version [16, Appendix C.1].

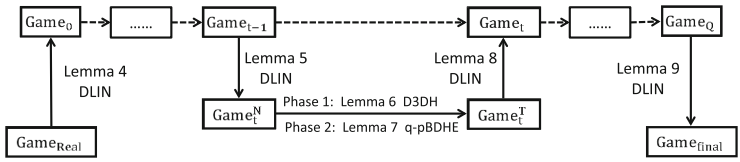

Fig. 1. Lemmas 4, 5, 8, and 9 rely on the subspace assumption, which is implied by the DLIN assumption, Lemma 6 relies on the D3DH assumption, and Lemma 7 relies on the source group q-parallel BDHE assumption.

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

Proof. The argument for security of $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}_{N+1}}$ is very straightforward since an encryption to index $N+1 = (n+1, 1)$ 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 $(n+1, 1)$. Since for all $i = 1$ to n, the values of T_i contain no information about the message, the bit b is perfectly hidden and $\mathsf{MH}^{\mathsf{A}}_{N+1}\mathsf{Adv}_{\mathcal{A}} = 0$.

Theorem 4. *Suppose that the D3DH assumption and the DLIN assumption hold. Then no PPT adversary can win* $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$ *with non-negligible advantage.*

Proof. Theorem 4 follows Lemmas 1 and 2 below.

Lemma 1. *Suppose that the D3DH assumption holds. Then for* $\bar{j} < n$ *no PPT adversary can distinguish between an encryption to* $(\bar{i}, \bar{j})$ *and* $(\bar{i}, \bar{j}+1)$ *in* $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$ *with non-negligible advantage.*

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

Case I: In Key Query phase, $\mathcal{A}$ will not submit $((\bar{i}, \bar{j}), S_{(\bar{i},\bar{j})})$ for some attribute set $S_{(\bar{i},\bar{j})}$ to query the corresponding private key. In Challenge phase, $\mathcal{A}$ submits a message M and a non-empty attribute set S^*. There is not any restriction on S^*.

Case II: In Key Query phase, $\mathcal{A}$ will submit $((\bar{i}, \bar{j}), S_{(\bar{i},\bar{j})})$ for some attribute set $S_{(\bar{i},\bar{j})}$ to query the corresponding private key. In Challenge phase, $\mathcal{A}$ submits a message M and a non-empty attribute set S^* with the restriction that the corresponding strictest access policy $\mathbb{A}_{S^*}$ is not satisfied by $S_{(\bar{i},\bar{j})}$ (i.e., $S^* \setminus S_{(\bar{i},\bar{j})} \neq \emptyset$).

The simulation for **Case I** is very similar to that of [5] 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 $\boldsymbol{Z}_i^{t_i}$ (in $\boldsymbol{Q}_i'$) and $\boldsymbol{Q}_i'' = g^{t_i(\boldsymbol{b}_1+\boldsymbol{b}_2)}$ in the ciphertext, and $\boldsymbol{Y}_j$ in the public parameter, our particular construction guarantees that $\mathcal{B}$ can successfully finish the simulation with probability $|S^* \setminus S_{(\bar{i},\bar{j})}|/|\mathcal{U}|$, which is at least $1/|\mathcal{U}|$ since $S^* \setminus S_{(\bar{i},\bar{j})} \neq \emptyset$. As of the fully secure CP-ABE schemes in [10–13,19], we assume that the size of attribute universe (i.e. $|\mathcal{U}|$) is polynomial in the security parameter λ, so that a degradation of $O(1/|\mathcal{U}|)$ in the security reduction is acceptable. The proof details of Lemma 1 can be found in the full version [16, Appendix C.2].

Lemma 2. *Suppose the D3DH assumption and the DLIN assumption hold. Then for any* $1 \leq \bar{i} \leq n$ *no PPT adversary can distinguish between an encryption to* $(\bar{i}, n)$ *and* $(\bar{i}+1, 1)$ *in* $\mathsf{Game}_{\mathsf{IH}}^{\mathsf{A}}$ *with non-negligible advantage.*

Proof. The proof of this lemma follows from a series of lemmas that establish the indistinguishability of the following games, where "less-than row" implies the corresponding $\boldsymbol{v}_i$ is randomly chosen from $\mathbb{Z}_p^3$ and T_i is a random element (i.e. $T_i = e(g,g)^{\hat{s}_i}$), "target row" implies the corresponding $\boldsymbol{v}_i$ is randomly chosen from $\mathbb{Z}_p^3$ and T_i is well-formed, and "greater-than row" implies the corresponding $\boldsymbol{v}_i$ is randomly chosen from $span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$ and T_i is well-formed.

- H_1: Encrypt to column n, row $\bar{i}$ is the target row, row $\bar{i}+1$ is the greater-than row.
- H_2: Encrypt to column $n+1$, row $\bar{i}$ is the target row, row $\bar{i}+1$ is the greater-than row.
- H_3: Encrypt to column $n+1$, row $\bar{i}$ is the less-than row, row $\bar{i}+1$ is the greater-than row (no target row).
- H_4: Encrypt to column 1, row $\bar{i}$ is the less-than row, row $\bar{i}+1$ is the greater-than row (no target row).
- H_5: Encrypt to column 1, row $\bar{i}$ is the less-than row, row $\bar{i}+1$ is the target row.

It can be observed that game H_1 corresponds to the encryption being done to $(\bar{i}, n)$ and game H_5 corresponds to encryption to $(\bar{i}+1, 1)$. As shown in Fig. 2, we use a series of lemmas, i.e. Lemmas 10, 11, 12, and 13, to prove the indistinguishability of the games H_1 and H_5. The details of these lemmas and their proofs can be found in the full version [16, Appendix C.3].

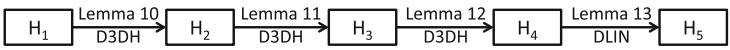

Fig. 2. Lemmas 10, 11, and 12 rely on the D3DH assumption, and Lemma 13 relies on the DLIN assumption.

4 Conclusion

In this paper, we proposed a new Augmented CP-ABE construction on prime order groups, and proved its message-hiding and index-hiding properties in the standard model. This implies the first CP-ABE that simultaneously achieves (1) fully collusion-resistant blackbox traceability in the standard model, (2) full security in the standard model, and (3) on prime order groups. The scheme is highly expressive in supporting any monotonic access structures, and as a fully collusion-resistant blackbox traceable CP-ABE scheme, it achieves the most efficient level to date, with the overhead in $O(\sqrt{N})$ only.

References

1. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy, pp. 321–334 (2007)
2. Cheung, L., Newport, C.C.: Provably secure ciphertext policy ABE. In: ACM Conference on Computer and Communications Security, pp. 456–465 (2007)
3. Deng, H., Wu, Q., Qin, B., Mao, J., Liu, X., Zhang, L., Shi, W.: Who is touching my cloud. In: Kutyłowski, M., Vaidya, J. (eds.) ICAIS 2014, Part I. LNCS, vol. 8712, pp. 362–379. Springer, Heidelberg (2014)
4. Freeman, D.M.: Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 44–61. Springer, Heidelberg (2010)
5. Garg, S., Kumarasubramanian, A., Sahai, A., Waters, B.: Building efficient fully collusion-resilient traitor tracing and revocation schemes. In: ACM Conference on Computer and Communications Security, pp. 121–130 (2010)
6. 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 2008, Part II. LNCS, vol. 5126, pp. 579–591. Springer, Heidelberg (2008)
7. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM Conference on Computer and Communications Security, pp. 89–98 (2006)
8. Herranz, J., Laguillaumie, F., Ràfols, C.: Constant size ciphertexts in threshold attribute-based encryption. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 19–34. Springer, Heidelberg (2010)
9. Lewko, A.B.: Tools for simulating features of composite order bilinear groups in the prime order setting. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 318–335. Springer, Heidelberg (2012)
10. 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 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010)
11. 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 2012. LNCS, vol. 7417, pp. 180–198. Springer, Heidelberg (2012)
12. Lewko, A.B., Waters, B.: New proof methods for attribute-based encryption: achieving full security through selective techniques. IACR Cryptology ePrint Archive 2012: 326 (2012)

13. Liu, Z., Cao, Z., Wong, D.S.: Blackbox traceable CP-ABE: how to catch people leaking their keys by selling decryption devices on ebay. In: ACM Conference on Computer and Communications Security, pp. 475–486 (2013)
14. Liu, Z., Cao, Z., Wong, D.S.: White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures. IEEE Trans. Inf. Forensics Secur. **8**(1), 76–88 (2013)
15. Liu, Z., Wong, D.S.: Practical attribute based encryption: Traitor tracing, revocation, and large universe. IACR Cryptology ePrint Archive, 2014:616 (2014)
16. Liu, Z., Wong, D.S.: Traceable CP-ABE on prime order groups: Fully secure and fully collusion-resistant blackbox traceable. IACR Cryptology ePrint Archive 2015:850 (2015)
17. Okamoto, T., Takashima, K.: Homomorphic encryption and signatures from vector decomposition. In: Galbraith, S.D., Paterson, K.G. (eds.) Pairing 2008. LNCS, vol. 5209, pp. 57–74. Springer, Heidelberg (2008)
18. Okamoto, T., Takashima, K.: Hierarchical predicate encryption for inner-products. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 214–231. Springer, Heidelberg (2009)
19. Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 191–208. Springer, Heidelberg (2010)
20. Rouselakis, Y., Waters, B.: Practical constructions and new proof methods for large universe attribute-based encryption. In: ACM Conference on Computer and Communications Security, pp. 463–474 (2013)
21. Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011)