Practical Attribute Based Encryption: Traitor Tracing, Revocation, and Large Universe

Zhen Liu and Duncan S. Wong

City University of Hong Kong, Hong Kong SAR, China.
zhenliu7@cityu.edu.hk, duncan@cityu.edu.hk

Abstract. In Ciphertext-Policy Attribute-Based Encryption (CP-ABE), a user's decryption key is associated with attributes which in general are not related to the user's identity, and the same set of attributes could be shared between multiple users. From the decryption key, if the user 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 addition, being able to revoke compromised keys is also an important step towards practicality, and for scalability, the scheme should support an exponentially large number of attributes. However, none of the existing traceable CP-ABE schemes supports revocation or large attribute universe. In this paper, we construct the first practical CP-ABE which possesses these three important properties: (1) blackbox traceability, (2) revocation, and (3) supporting large universe. When compared with the latest 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, and attains the same security level, namely, the fully collusion-resistant traceability against policy-specific decryption blackbox, which is proven against selective adversaries in the standard model. The scheme also supports large attribute universe, and attributes do not need to be pre-specified during the system setup. It is highly expressive and can take any monotonic access structures as ciphertext policies.
We also present the analogous results in the Key-Policy Attribute-Based Encryption (KP-ABE) setting, where users' description keys are described by access policies and ciphertexts are associated with attributes. We construct the first practical KP-ABE which possesses the three important properties: (1) blackbox traceability, (2) revocation, and (3) supporting large universe. The scheme is highly expressive and can take any monotonic access structures as key policies, and is efficient, namely, enjoys the sub-linear overhead of $O(\sqrt{N})$ while supporting fully collusion-resistant blackbox traceability and revocation, and does not need to pre-specify the attributes during the system setup. The scheme is proven selectively secure in the standard model.

Keywords: Attribute-Based Encryption, Traitor Tracing, Revocation, Large Attribute Universe

1 Introduction

In some emerging applications such as user-side encrypted cloud storage, users may store encrypted data on a public untrusted cloud and let other users who have eligible credentials decrypt and access the data. The decryption credentials could be based on the users' roles and do not have to be their identities. For example, a user Alice wants to encrypt some documents, upload to the cloud, and let all PhD students and alumni in the Department of Mathematics download and decrypt, while she does not have the identities of all the eligible receivers, and the set of eligible receivers could also be dynamic. Intuitively, Alice is to encrypt the documents under "(Mathematics AND (PhD Student OR Alumni))", which is an *access policy* defined over descriptive *attributes*, so that only those receivers whose attributes satisfy this policy can decrypt. Traditional public key encryption, identity-based encryption (e.g. [5]) and broadcast encryption (e.g. [23]) are user-specific, where Alice has to know the exact identities of all the desired receivers so that she can use

the corresponding public key/identity/index to encrypt the documents, and thus are not suitable. *Attribute-Based Encryption* (ABE), introduced by Sahai and Waters [28], provides a solution to this type of applications. In a Ciphertext-Policy ABE (CP-ABE) [11,2] scheme[1], each user possesses a set of attributes/credentials and a secret key, which corresponds to these credentials, the encrypting party can encrypt the data using an access policy (e.g. a Boolean formula) on attributes, and a user can decrypt if and only if the user's attributes satisfy the policy.

CP-ABE has attracted much attention in recent years, and among the recently proposed schemes [2,8,10,30,16,24,12,18], one of the state-of-the-art works is due to Lewko and Waters [18,19]. Their scheme achieves high expressivity (i.e. can take any monotonic access structures as ciphertext policies), and is provably secure against adaptive adversaries in the standard model. The scheme is also efficient and removes the one-use restriction that other comparable schemes have [16,24]. As of the current Public Key Infrastructure which mandates the capabilities of key generation, revocation, and certified binding between identities and public keys, before the CP-ABE being able to deploy in practice, we should provision a practical CP-ABE scheme with three important features: (1) traceability, (2) revocation, and (3) large universe. Very recently, a handful of research works have been done on each one of these while the fundamental open problem remains is the existence of an efficient scheme which supports these three features at once.

Traceability / Traitor Tracing. Access policies in CP-ABE do not have to contain any receivers' identities, and more commonly, a CP-ABE policy is role-based and attributes are *shared* between multiple users. In practice, a malicious user, with attributes shared with multiple other users, might leak a decryption blackbox/device, which is made of the user's decryption key, for the purpose of financial gain or some other forms of incentives, as the malicious user has little risk of being identified out of all the users who can build a decryption blackbox with identical decryption capability. Being able to identify this malicious user is crucial towards the practicality of a CP-ABE system.

Given a well-formed decryption key, if the *tracing algorithm* of a CP-ABE scheme can find out the malicious user who created the key from his/her original key, the scheme is called Whitebox Traceable CP-ABE [21]. Given a decryption blackbox, while the decryption key and even the decryption algorithm could be hidden inside the blackbox, if the *tracing algorithm* can still find out the traitor whose key has been used in constructing the blackbox, the scheme is called Blackbox Traceable CP-ABE [20]. In this stronger blackbox traceability notion, there are two types of blackboxes: key-like decryption blackbox and policy-specific decryption blackbox. A key-like decryption blackbox has an attribute set associated and can decrypt encrypted messages with policies being satisfied by the attribute set. A policy-specific decryption blackbox has a policy associated and can decrypt encrypted messages with the same policy. According to [20], it is believed that traceability against a policy-specific decryption blackbox is no easier to achieve than that against a key-like decryption blackbox. In fact, the CP-ABE scheme proposed by Liu et al. in [20] is conjectured to be traceable against policy-specific decryption blackbox in the standard model with selective adversaries. On the other side, the scheme is highly expressive and achieves the most efficient level to date, i.e. the overhead for traceability is in $O(\sqrt{N})$, where N is the number of users in the system.

Revocation. For any encryption systems that involve many users, private keys might get compromised, users might leave or be removed from the systems. When any of these happens, the corresponding user keys should be revoked. In the literature, several revocation mechanisms have been proposed in the context of CP-ABE. In [31], Yu et al. proposed a mechanism which requires

[1] Here we focus on CP-ABE, while skipping the corresponding discussions about Key-Policy ABE.

a semi-trusted proxy server to be online. In [27][2], Sahai et al. proposed an *indirect* revocation mechanism, which requires an authority to periodically broadcast a key update information so that only the non-revoked users can update their keys and continue to decrypt messages. In [1], Attrapadung and Imai proposed a *direct* revocation mechanism, which allows a revocation list to be specified directly during encryption so that the resulting ciphertext cannot be decrypted by any decryption key which is in the revocation list even though the associated attribute set of the key satisfies the ciphertext policy. The direct revocation mechanism does not need any periodic key updates that an indirect revocation mechanism requires. It does not affect any non-revoked users either. While for indirect revocation mechanism, defining an appropriate time period for key updates could be a difficult task in practice: if the time period is too long, the revocation cannot take effect in time, on the other side, if the time slot is too short, the frequent key updates could become an expensive overhead for the system. In direct revocation, a system-wide revocation list could be made public and revocation could be taken into effect promptly as the revocation list could be updated immediately once a key is revoked. In this paper, we focus on achieving direct revocation in CP-ABE.

Large Attribute Universe. In most CP-ABE schemes, the size of the attribute universe is polynomially bounded in the security parameter, and the attributes have to be fixed during the system setup. In a large universe CP-ABE, the attribute universe can be exponentially large, any string can be used as an attribute, and attributes do not need to be pre-specified during setup. Although "somewhat" large universe CP-ABE schemes have been proposed or discussed previously [30,16,1,25], as explained by Rouselakis and Waters [26], limitations exist. The first "truly" large universe CP-ABE construction, in which there is no restriction on ciphertext policies or attributes associated with the decryption keys, was proposed in [26].

1.1 Our Results

We propose the first practical CP-ABE scheme that simultaneously supports (1) traceability against policy-specific decryption blackbox, (2) (direct) revocation and (3) "truly" large attribute universe. We show that the scheme's traceability is fully collusion-resistant, that is, the number of colluding users in constructing a decryption blackbox is not limited and can be arbitrary. Furthermore, the traceability is public, that is, anyone can run the tracing algorithm. The scheme is also highly expressive that allows any monotonic access structures to be the ciphertext policies.

The scheme is proven selectively secure and traceable in the standard model. This is comparable to the traceable CP-ABE against policy-specific decryption blackbox with traceability conjectured to be selective [20] and also to the security of the "truly" large universe CP-ABE [26]. The selective security is indeed a weakness when compared with the full security of [18,20], but as discussed in [26], selective security is still a meaningful notion and can be a reasonable trade off for performance in some circumstances. Furthermore, in light of the proof method of [18] that achieves full security through selective techniques, we can see that developing selectively secure schemes could be an important stepping stone towards building fully secure ones.

Table 1 compares this new scheme with the representative results in conventional CP-ABE [18], blackbox traceable CP-ABE [20], revocable CP-ABE [1], and "truly" large universe CP-ABE [26], in terms of features (i.e. blackbox traceability, revocation and large universe) and performance.

[2] Note that in this paper we focus on the the conventional revocation, which is to prevent a compromised or revoked user from decrypting newly encrypted messages. In [27], revoking access on previously encrypted data is also considered.

[1] [2]	Blackbox Traceability	Revocation	Large Universe	Public Key Size	Ciphertext Size	Private Key Size	Pairings in Decryption
[18,19] [3]	×	×	×	$14+6\|\mathcal{U}\|$	$7+6l$	$6+6\|S\|$	$9+6\|I\|$
[1, Sec. 5.1]	×	√	∂ [4]	$2N+2+m+l_m$ [4]	$3+l$	$2+\|S\|$	$3+2\|I\|$
[1, Sec. 5.2]	×	√	∂ [4]	$7+m+l_m$ [4]	$2+l+2\|R\|$ [4]	$4+\|S\|$	$1+2\|I\|+2\|R\|$ [4]
[26]	×	×	√	6	$2+3l$	$2+2\|S\|$	$1+3\|I\|$
[20] [5]	√	×	×	$3+4\sqrt{N}+\|\mathcal{U}\|$	$17\sqrt{N}+2l$	$4+\|S\|$	$10+2\|I\|$
this work	√	√	√	$5+5\sqrt{N}$	$16\sqrt{N}+3l$	$2+\sqrt{N}+2\|S\|$	$9+3\|I\|$

[1] All the five schemes are highly expressive, i.e. supporting any monotonic access structures.

[2] Let N be the number of users in the system, $|\mathcal{U}|$ the size of the attribute universe, l the number of rows of the LSSS matrix for an access policy, $|S|$ the size of the attribute set of a decryption key, and $|I|$ the number of attributes for a decryption key to satisfy a ciphertext policy.

[3] [19], as the full version of [18], provides a prime order construction, and the efficiency evaluation here is based on it.

[4] The CP-ABE schemes in [1] are not "truly" large universe, as some limitations are imposed and some corresponding parameters have to be fixed during the setup. Let m be the maximum size of an attribute set associated with a key, l_m the maximum number of rows in the LSSS matrix of a policy, and $|R|$ the number of revoked users in a revocation list R.

[5] The construction in [20] is on composite order groups, and the order is the product of three large primes, and the efficiency evaluation is based on the composite order groups. As a result, the *actual* sizes of public key and ciphertext in [20] is larger than that of this work, and the encryption and decryption in [20] are slower than that of this work.

Table 1. Features and Efficiency Comparison

The scheme's overhead is in $O(\sqrt{N})$, where N is the number of users in a system. This might be a concern, but we stress that for fully collusion-resistant blackbox traceable CP-ABE, such a sub-linear overhead is the most efficient one to date. Furthermore, when compared with the existing fully collusion-resistant blackbox traceable CP-ABE scheme in [20], at the cost of $\sqrt{N}$ additional elements in private key, our construction achieves revocation and "truly" large universe. For achieving better performance, this new scheme is constructed on prime order groups, rather than composite order groups, as it has been showed (e.g. in [9,15]) that constructions on composite order groups will result in significant loss of efficiency.

Paper Outline. In Sec. 2, by following the definitions in [18,20,26], we propose a definition for CP-ABE supporting traceability against policy-specific decryption blackbox, direct revocation and large attribute universe. As of [20], the definition is 'functional', namely each decryption key is uniquely indexed by $k \in \{1, \ldots, N\}$ (N is the number of users in the system) and given a policy-specific decryption blackbox, the tracing algorithm Trace can return the index k of a decryption key which has been used for building the decryption blackbox. On direct revocation, in our definition, the $\mathsf{Encrypt}$ algorithm takes a revocation list $R \subseteq \{1, \ldots, N\}$ as an additional input so that a message is encrypted under the (revocation list, access policy) pair $(R, \mathbb{A})$ would only allow users whose (index, attribute set) pair (k, S) satisfies $(k \notin R) \wedge (S \; satisfies \; \mathbb{A})$ to decrypt.

On the construction, we refer to the 'functional' CP-ABE in Sec. 2 as Revocable CP-ABE (or R-CP-ABE for short), then as analogous to the approach in [20], we extend the R-CP-ABE to a primitive called Augmented R-CP-ABE (or AugR-CP-ABE for short), which will lastly be transformed to a policy-specific blackbox traceable R-CP-ABE. More specifically, in Sec. 3, we define the encryption algorithm of AugR-CP-ABE as $\mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, R, \mathbb{A}, \bar{k})$ which takes one more parameter $\bar{k} \in \{1, \ldots, N+1\}$ than the original one in R-CP-ABE. This also changes the decryption criteria in AugR-CP-ABE in such a way that an encrypted message can be recovered

using a decryption key $\mathsf{SK}_{k,S}$, which is identified by index $k \in \{1, \ldots, N\}$ and associated with an attribute set S, only if $(k \notin R) \wedge (S\ satisfies\ \mathbb{A}) \wedge (k \geq \bar{k})$. Also, we formalize a message-hiding game and an index-hiding game, and show that an AugR-CP-ABE scheme satisfying the message-hiding and the (selective) index-hiding can be transformed to a (selectively) secure R-CP-ABE with (selective) policy-specific blackbox traceability.

In Sec. 4, we propose a *large universe* AugR-CP-ABE, and show that it is message-hiding and selective index-hiding in the standard model. Combining it with the results in Sec. 3, we obtain a large universe R-CP-ABE construction, which is efficient (with overhead size in $O(\sqrt{N})$), highly expressive (supporting any monotonic access structures as policies), selectively secure and selectively policy-specific blackbox traceable in the standard model.

To construct the AugR-CP-ABE, we borrow ideas from some existing works, such as the CP-ABE constructions in [20,26] and Trace&Revoke schemes in [9]. However, the combination is not trivial and may result in inefficient or insecure systems. In particular, besides achieving the important features for practicality, such as traitor tracing, revocation, large universe, high expressivity and efficiency, we achieve provable security and traceability in the standard model. As we will discuss later in Sec. 4, proving the blackbox traceability while supporting the large attribute universe is one of the most challenging tasks in this work. As we can see, the proof techniques for blackbox traceability in [20] are no longer applicable for large universe, while that for large universe in [26] are only for confidentiality rather than for blackbox traceability.

Following a similar route, we also present the analogous results in Key-Policy ABE setting, as shown in Sec. 5.

2 Revocable CP-ABE and Blackbox Traceability

In this section, we define Revocable CP-ABE (or R-CP-ABE for short) and its security, which are based on conventional (non-traceable, non-revocable) CP-ABE (e.g. [18]). Similar to the traceable CP-ABE in [20], in our 'functional' definition, we explicitly assign and identify users using unique indices. Then we formalize traceability against policy-specific decryption blackbox on R-CP-ABE.

2.1 Revocable CP-ABE

Given a positive integer n, let $[n]$ be the set $\{1, 2, \ldots, n\}$. A Revocable Ciphertext-Policy Attribute-Based Encryption (R-CP-ABE) scheme consists of four algorithms:

- $\mathsf{Setup}(\lambda, N) \to (\mathsf{PP}, \mathsf{MSK})$. The algorithm takes as input a security parameter λ and the number of users in the system N, runs in polynomial time in λ, and outputs a public parameter PP and a master secret key MSK. We assume that PP contains the description of the attribute universe $\mathcal{U}$[3].
- $\mathsf{KeyGen}(\mathsf{PP}, \mathsf{MSK}, S) \to \mathsf{SK}_{k,S}$. The algorithm takes as input the public parameter PP, the master secret key MSK, and an attribute set S, and outputs a private decryption key $\mathsf{SK}_{k,S}$, which is assigned and identified by a unique index $k \in [N]$.
- $\mathsf{Encrypt}(\mathsf{PP}, M, R, \mathbb{A}) \to CT_{R,\mathbb{A}}$. The algorithm takes as input PP, a message M, a revocation list $R \subseteq [N]$, and an access policy $\mathbb{A}$ over $\mathcal{U}$, and outputs a ciphertext $CT_{R,\mathbb{A}}$ such that only users whose indices are not revoked by R and attributes satisfy $\mathbb{A}$ can recover M. $(R, \mathbb{A})$ is included in $CT_{R,\mathbb{A}}$.

[3] For large universe and also in our work, the attribute universe depends only on the size of the underlying group $\mathbb{G}$, which depends on λ and the group generation algorithm.

$\mathsf{Decrypt}(\mathsf{PP}, CT_{R,\mathbb{A}}, \mathsf{SK}_{k,S}) \rightarrow M$ or $\perp$. The algorithm takes as input PP, a ciphertext $CT_{R,\mathbb{A}}$, and a decryption key $\mathsf{SK}_{k,S}$. If $(k \in [N] \setminus R)$ AND $(S\ satisfies\ \mathbb{A})$, the algorithm outputs a message M, otherwise it outputs $\perp$ indicating the failure of decryption.

The security of the R-CP-ABE is defined as follows.

$\mathsf{Game}_{\mathsf{MH}}$. This message-hiding game is defined between a challenger and an adversary $\mathcal{A}$.

Setup. The challenger runs $\mathsf{Setup}(\lambda, N)$ and gives the public parameter PP to $\mathcal{A}$.
Phase 1. For $i = 1$ to Q_1, $\mathcal{A}$ adaptively submits (index, attribute set) pair (k_i, S_{k_i}). The challenger responds with $\mathsf{SK}_{k_i,S_{k_i}}$.
Challenge. $\mathcal{A}$ submits two equal-length messages M_0, M_1 and a (revocation list, access policy) pair $(R^*, \mathbb{A}^*)$. The challenger flips a random coin $b \in \{0,1\}$, and sends $CT_{R^*,\mathbb{A}^*} \leftarrow \mathsf{Encrypt}(\mathsf{PP}, M_b, R^*, \mathbb{A}^*)$ to $\mathcal{A}$.
Phase 2. For $i = Q_1 + 1$ to Q, $\mathcal{A}$ adaptively submits (index, attribute set) pair (k_i, S_{k_i}). The challenger responds with $\mathsf{SK}_{k_i,S_{k_i}}$.
Guess. $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ for b.

$\mathcal{A}$ wins the game if $b' = b$ under the **restriction** that none of the queried $\{(k_i, S_{k_i})\}_{i=1}^{Q}$ can satisfy $(k_i \in [N] \setminus R^*)$ AND $(S_{k_i}\ satisfies\ \mathbb{A}^*)$. The advantage of $\mathcal{A}$ is defined as $\mathsf{MHAdv}_{\mathcal{A}} = |\Pr[b' = b] - \frac{1}{2}|$.

Definition 1. *An N-user R-CP-ABE scheme is secure if for all probabilistic polynomial time (PPT) adversaries $\mathcal{A}$, $\mathsf{MHAdv}_{\mathcal{A}}$ is negligible in λ.*

We say that an N-user R-CP-ABE scheme is *selectively* secure if we add an **Init** stage before **Setup** where the adversary commits to the access policy $\mathbb{A}^*$.

It is worth noticing that: (1) although the KeyGen algorithm is responsible for determining/assigning the index of each user's decryption key, to capture the security that an adversary can adaptively choose decryption keys to corrupt, we allow $\mathcal{A}$ to specify the index when querying for a key, i.e., for $i = 1$ to Q, $\mathcal{A}$ submits pairs of (k_i, S_{k_i}) for decryption keys with attribute sets corresponding to S_{k_i}, where $Q \leq N$, $k_i \in [N]$, and $k_i \neq k_j\ \forall 1 \leq i \neq j \leq Q$ (this is to guarantee that each user/key can be *uniquely* identified by an index); and (2) for $k_i \neq k_j$ we do not require $S_{k_i} \neq S_{k_j}$, i.e., different users/keys may have the same attribute set.

Remark: (1) The R-CP-ABE defined above extends the conventional definition for non-revocable CP-ABE [18], where the revocation list R is always empty. (2) For traceability, we explicitly assign a unique index to each user's decryption key. Predefining the number of users N in the system is indeed a weakness but is also a necessary price to pay for achieving blackbox traceability, but we stress that in practice, this should not incur any noticeable concern, and in fact, all the existing blackbox traceable systems (e.g. [6,7,9,14,20]) have the same setting. (3) When the revocation list R needs an update due to, for example, some decryption keys being compromised or some users leaving the system, the updated R needs to be disseminated to encrypting parties. In practice, this can be done in a similar way to the certificate revocation list distribution in the existing Public Key Infrastructure, namely an authority may update R, and publish it together with the authority's signature generated on it. There are many ways for the encrypting parties to obtain a copy of the updated R, for example, via RSS feeds. (4) From the view of the public, R is just a set of numbers (in $[N]$). These numbers (or indices) do not have to provide any information on the corresponding users, in fact, besides the authority who runs KeyGen, each user only knows his/her own index. Also, encrypting parties do not need to know the indices of any users in order to encrypt but only the

access policies. Although associating a revocation list with a ciphertext might make the resulting CP-ABE look less purely attribute-based, it does not undermine the capability of CP-ABE, that is, enabling fine-grained access control on encrypted messages.

2.2 Blackbox Traceability

A policy-specific decryption blackbox $\mathcal{D}$ in the setting of R-CP-ABE is viewed as a probabilistic circuit that can decrypt ciphertexts generated under some specific pair of revocation list and access policy. *In particular, a policy-specific decryption blackbox $\mathcal{D}$ is described by a (revocation list, access policy) pair $(R_{\mathcal{D}}, \mathbb{A}_{\mathcal{D}})$ and a non-negligible probability value ϵ (i.e. $0 < \epsilon \leq 1$ is polynomial in λ), and this blackbox $\mathcal{D}$ can decrypt ciphertexts generated under $(R_{\mathcal{D}}, \mathbb{A}_{\mathcal{D}})$ with probability at least ϵ.* Such a blackbox can reflect most practical scenarios, which include the key-like decryption blackbox for sale and decryption blackbox "found in the wild", which are discussed in [20]. In particular, once a blackbox is found being able to decrypt ciphertexts (regardless of how this is found, for example, an explicit description of the blackbox's decryption ability is given, or the law enforcement agency finds some clue), we can regard it as a policy-specific decryption blackbox with the corresponding (revocation list, access policy) pair (which is associated to the ciphertext). And for a decryption blackbox, if multiple (revocation list, access policy) pairs are found that corresponding ciphertexts can be decrypted by it with non-negligible probability, we can regard the blackbox as multiple policy-specific decryption blackboxes, each with a different (revocation list, access policy) pair.

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

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

The following tracing game captures the notion of fully collusion-resistant traceability against policy-specific decryption blackbox. In the game, the adversary targets to build a decryption blackbox $\mathcal{D}$ that can decrypt ciphertexts under some (revocation list, access policy) pair $(R_{\mathcal{D}}, \mathbb{A}_{\mathcal{D}})$. The tracing algorithm, on the other side, is designed to extract the index of at least one of the malicious users whose decryption keys have been used for constructing $\mathcal{D}$.

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

Setup. The challenger runs $\mathsf{Setup}(\lambda, N)$ and gives the public parameter PP to $\mathcal{A}$.
Key Query. For $i = 1$ to Q, $\mathcal{A}$ adaptively submits (index, attribute set) pair (k_i, S_{k_i}). The challenger responds with $\mathsf{SK}_{k_i, S_{k_i}}$.
Decryption Blackbox Generation. $\mathcal{A}$ outputs a decryption blackbox $\mathcal{D}$ associated with a (revocation list, access policy) pair $(R_{\mathcal{D}}, \mathbb{A}_{\mathcal{D}})$ and a non-negligible probability value ϵ.
Tracing. The challenger runs $\mathsf{Trace}^{\mathcal{D}}(\mathsf{PP}, R_{\mathcal{D}}, \mathbb{A}_{\mathcal{D}}, \epsilon)$ to obtain an index set $\mathbb{K}_T \subseteq [N]$.

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

1. $\Pr[\mathcal{D}(\mathsf{Encrypt}(\mathsf{PP}, M, R_{\mathcal{D}}, \mathbb{A}_{\mathcal{D}})) = M] \geq \epsilon$, where the probability is taken over the random choices of message M and the random coins of $\mathcal{D}$. A decryption blackbox satisfying this condition is said to be a *useful policy-specific decryption blackbox*.

2. $\mathbb{K}_T = \emptyset$, or $\mathbb{K}_T \not\subseteq \mathbb{K}_{\mathcal{D}}$, or $((k_t \in R_{\mathcal{D}})$ OR $(S_{k_t}$ *does not satisfy* $\mathbb{A}_{\mathcal{D}})\ \forall k_t \in \mathbb{K}_T)$.

We denote by $\mathsf{TRAdv}_{\mathcal{A}}$ the probability that $\mathcal{A}$ wins.

For a useful policy-specific decryption blackbox $\mathcal{D}$, the traced $\mathbb{K}_T$ must satisfy $(\mathbb{K}_T \neq \emptyset) \wedge (\mathbb{K}_T \subseteq \mathbb{K}_{\mathcal{D}}) \wedge (\exists k_t \in \mathbb{K}_T\ s.t.\ (k_t \in [N] \setminus R_{\mathcal{D}})$ AND $(S_{k_t}\ satisfies\ \mathbb{A}_{\mathcal{D}}))$. (1) $(\mathbb{K}_T \neq \emptyset) \wedge (\mathbb{K}_T \subseteq \mathbb{K}_{\mathcal{D}})$ captures the preliminary traceability that the tracing algorithm can extract at least one malicious user and the coalition of malicious users cannot frame any innocent user. (2) $(\exists k_t \in \mathbb{K}_T\ s.t.\ (k_t \in [N] \setminus R_{\mathcal{D}})$ AND $(S_{k_t}\ satisfies\ \mathbb{A}_{\mathcal{D}}))$ captures the *strong traceability* that the tracing algorithm can extract at least one malicious user whose decryption key enables $\mathcal{D}$ to have the decryption ability corresponding to $(R_{\mathcal{D}}, \mathbb{A}_{\mathcal{D}})$, i.e. whose index is not in $R_{\mathcal{D}}$ and whose attribute set satisfies $\mathbb{A}_{\mathcal{D}}$. Strong traceability is desirable in practice, since it can defend against attacks where colluding traitors may build $\mathcal{D}$ in a smart manner so that $\mathcal{D}$ will be traced to only a user whose index is in $R_{\mathcal{D}}$ or whose attributes do not satisfy $\mathbb{A}_{\mathcal{D}}$, which should not happen for a secure R-CP-ABE. We refer to [14,20] on why strong traceability is desirable. Note that, as of [6,7,9,14,20], we are modeling a stateless (resettable) decryption blackbox – such a blackbox is just an oracle and maintains no state between activations. Also note that we are modeling public traceability, namely, the Trace algorithm does not need any secrets and anyone can perform the tracing from the public parameter only.

Definition 2. *An N-user R-CP-ABE scheme is traceable against policy-specific decryption blackbox if for all PPT adversaries $\mathcal{A}$, $\mathsf{TRAdv}_{\mathcal{A}}$ is negligible in λ.*

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

In the traceable CP-ABE of [20], given a decryption blackbox, it is guaranteed that at least one decryption key in the blackbox will be traced. But in the traceable R-CP-ABE above, it is possible to trace *all the active decryption keys* in the blackbox. In particular, given a decryption blackbox $\mathcal{D}$ described by $(R_{\mathcal{D}}, \mathbb{A}_{\mathcal{D}})$ and non-negligible probability ϵ, we can run Trace to obtain an index set $\mathbb{K}_T$ so that $(\mathbb{K}_T \neq \emptyset) \wedge (\mathbb{K}_T \subseteq \mathbb{K}_{\mathcal{D}}) \wedge (\exists k_t \in \mathbb{K}_T\ s.t.\ (k_t \in [N] \setminus R_{\mathcal{D}})$ AND $(S_{k_t}\ satisfies\ \mathbb{A}_{\mathcal{D}}))$. Then, we can set a new revocation list $R'_{\mathcal{D}} = R_{\mathcal{D}} \cup \{k_t \in \mathbb{K}_T \mid (k_t \in [N] \setminus R_{\mathcal{D}})$ AND $(S_{k_t}\ satisfies\ \mathbb{A}_{\mathcal{D}})\}$ and test whether $\mathcal{D}$ can decrypt ciphertexts under $(R'_{\mathcal{D}}, \mathbb{A}_{\mathcal{D}})$. If $\mathcal{D}$ can still decrypt the ciphertexts with non-negligible probability ϵ', we can run Trace on $(R'_{\mathcal{D}}, \mathbb{A}_{\mathcal{D}}, \epsilon')$ and obtain a new index set $\mathbb{K}'_T$, where $(\mathbb{K}'_T \neq \emptyset) \wedge (\mathbb{K}'_T \subseteq \mathbb{K}_{\mathcal{D}}) \wedge (\exists k_t \in \mathbb{K}'_T\ s.t.\ (k_t \in [N] \setminus R'_{\mathcal{D}})$ AND $(S_{k_t}\ satisfies\ \mathbb{A}_{\mathcal{D}}))$. By repeating this process, iteratively expanding the revocation list, until $\mathcal{D}$ can no longer decrypt the corresponding ciphertexts, we have finished finding out *all the active* malicious users of $\mathcal{D}$.

3 Augmented R-CP-ABE

As outlined in Sec. 1.1, we now define Augmented R-CP-ABE (or AugR-CP-ABE for short) from the R-CP-ABE above, formalize its security notions, then show that a secure AugR-CP-ABE can be transformed to a R-CP-ABE with blackbox traceability. In Sec. 4, we propose a concrete construction of AugR-CP-ABE.

3.1 Definitions

An AugR-CP-ABE scheme has four algorithms: $\mathsf{Setup}_{\mathsf{A}}$, $\mathsf{KeyGen}_{\mathsf{A}}$, $\mathsf{Encrypt}_{\mathsf{A}}$, and $\mathsf{Decrypt}_{\mathsf{A}}$. The setup and key generation algorithms are the same as that of R-CP-ABE. For the encryption algorithm, it takes one more parameter $\bar{k} \in [N+1]$ as input, and is defined as follows.

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

The decryption algorithm is also defined in the same way as that of R-CP-ABE. However, the correctness definition is changed to the following.

Correctness. For any $S \subseteq \mathcal{U}$, $k \in [N]$, $R \subseteq [N]$, $\mathbb{A}$ over $\mathcal{U}$, encryption index $\bar{k} \in [N+1]$, and M, suppose $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}_{\mathsf{A}}(\lambda, N)$, $\mathsf{SK}_{k,S} \leftarrow \mathsf{KeyGen}_{\mathsf{A}}(\mathsf{PP}, \mathsf{MSK}, S)$, $CT_{R,\mathbb{A}} \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, R, \mathbb{A}, \bar{k})$. If $(k \in [N] \setminus R) \wedge (S \text{ satisfies } \mathbb{A}) \wedge (k \geq \bar{k})$ then $\mathsf{Decrypt}_{\mathsf{A}}(\mathsf{PP}, CT_{R,\mathbb{A}}, \mathsf{SK}_{k,S}) = M$.

Note that during decryption, as long as $(k \in [N] \setminus R) \wedge (S \text{ satisfies } \mathbb{A})$, the decryption algorithm outputs a message, but only when $k \geq \bar{k}$, the output message is equal to the correct message, that is, if and only if $(k \in [N] \setminus R) \wedge (S \text{ satisfies } \mathbb{A}) \wedge (k \geq \bar{k})$, can $\mathsf{SK}_{k,S}$ correctly decrypt a ciphertext under $(R, \mathbb{A}, \bar{k})$. If we always set $\bar{k} = 1$, the functions of AugR-CP-ABE are identical to that of R-CP-ABE. In fact, the idea behind transforming an AugR-CP-ABE to a blackbox traceable R-CP-ABE, that we will show shortly, is to construct an AugR-CP-ABE with index-hiding property, and then always sets $\bar{k} = 1$ in normal encryption, while using $\bar{k} \in [N+1]$ to generate ciphertexts for tracing.

Security. We define the security of AugR-CP-ABE in two games. The first game is a **message-hiding game** and says that a ciphertext created using index $N+1$ is unreadable by anyone. The second game is an **index-hiding game** and captures the intuition that a ciphertext created using index $\bar{k}$ reveals no non-trivial information about $\bar{k}$.

$\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}}$. The **message-hiding game** $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}}$ is similar to $\mathsf{Game}_{\mathsf{MH}}$ except that during the **Challenge** phase, the challenge ciphertext is computed as $CT_{R^*,\mathbb{A}^*} \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M_b, R^*, \mathbb{A}^*, N+1)$, and the original **restriction** in $\mathsf{Game}_{\mathsf{MH}}$ no longer applies in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}}$. In particular, The **message-hiding game** $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}}$ proceeds as follows:

Setup. The challenger runs $\mathsf{Setup}_{\mathsf{A}}(\lambda, N)$ and gives the public parameter PP to $\mathcal{A}$.
Phase 1. For $i = 1$ to Q_1, $\mathcal{A}$ adaptively submits (index, attribute set) pair (k_i, S_{k_i}). The challenger responds with $\mathsf{SK}_{k_i,S_{k_i}}$.
Challenge. $\mathcal{A}$ submits two equal-length messages M_0, M_1 and a (revocation list, access policy) pair $(R^*, \mathbb{A}^*)$. The challenger flips a random coin $b \in \{0,1\}$, and sends $CT_{R^*,\mathbb{A}^*} \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M_b, R^*, \mathbb{A}^*, N+1)$ to $\mathcal{A}$.
Phase 2. For $i = Q_1 + 1$ to Q, $\mathcal{A}$ adaptively submits (index, attribute set) pair (k_i, S_{k_i}). The challenger responds with $\mathsf{SK}_{k_i,S_{k_i}}$.
Guess. $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ for b.

$\mathcal{A}$ wins the game if $b' = b$. The advantage of the adversary $\mathcal{A}$ in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}}$ is defined as $\mathsf{MH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}} = |\Pr[b' = b] - \frac{1}{2}|$.

Definition 3. *An N-user Augmented R-CP-ABE scheme is message-hiding in* $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}}$ *if for all PPT adversaries* $\mathcal{A}$ *the advantage* $\mathsf{MH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}$ *is negligible in* λ.

$\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$. In the **index-hiding game**, we require that, for any (revocation list, access policy) pair $(R^*, \mathbb{A}^*)$, an adversary cannot distinguish between a ciphertext under $(R^*, \mathbb{A}^*, \bar{k})$ and $(R^*, \mathbb{A}^*, \bar{k}+1)$ without a decryption key $\mathsf{SK}_{\bar{k},S_{\bar{k}}}$, where $(\bar{k} \in [N] \setminus R^*) \wedge (S_{\bar{k}}\ satisfies\ \mathbb{A}^*)$. 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 $\mathsf{Setup}_{\mathsf{A}}(\lambda, N)$ and gives the public parameter PP to $\mathcal{A}$.
Phase 1. For $i = 1$ to Q_1, $\mathcal{A}$ adaptively submits (index, attribute set) pair (k_i, S_{k_i}). The challenger responds with $\mathsf{SK}_{k_i,S_{k_i}}$.
Challenge. $\mathcal{A}$ submits a message M and a (revocation list, access policy) pair $(R^*, \mathbb{A}^*)$. The challenger flips a random coin $b \in \{0,1\}$, and sends $CT_{R^*,\mathbb{A}^*} \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, R^*, \mathbb{A}^*, \bar{k}+b)$ to $\mathcal{A}$.
Phase 2. For $i = Q_1 + 1$ to Q, $\mathcal{A}$ adaptively submits (index, attribute set) pair (k_i, S_{k_i}). The challenger responds with $\mathsf{SK}_{k_i,S_{k_i}}$.
Guess. $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ for b.

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

Definition 4. *An N-user Augmented R-CP-ABE scheme 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.*

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

3.2 The Reduction of Traceable R-CP-ABE to Augmented R-CP-ABE

Let $\Sigma_{\mathsf{A}} = (\mathsf{Setup}_{\mathsf{A}}, \mathsf{KeyGen}_{\mathsf{A}}, \mathsf{Encrypt}_{\mathsf{A}}, \mathsf{Decrypt}_{\mathsf{A}})$ be an AugR-CP-ABE, define $\mathsf{Encrypt}(\mathsf{PP}, M, R, \mathbb{A}) = \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, R, \mathbb{A}, 1)$, then $\Sigma = (\mathsf{Setup}_{\mathsf{A}}, \mathsf{KeyGen}_{\mathsf{A}}, \mathsf{Encrypt}, \mathsf{Decrypt}_{\mathsf{A}})$ is a R-CP-ABE derived from Σ_{A}. In the following, we show that if Σ_{A} is message-hiding and index-hiding, then Σ is secure (w.r.t. Def. 1). Furthermore, we propose a tracing algorithm Trace for Σ and show that if Σ_{A} is message-hiding and index-hiding, then Σ (equipped with Trace) is traceable (w.r.t. Def. 2).

3.2.1 R-CP-ABE Security

Theorem 1. *If Σ_{A} is message-hiding and index-hiding (resp. selectively index-hiding), then Σ is secure (resp. selectively secure).*

Proof. First we need a slightly more elaborate message-hiding game for Σ_{A}. In addition to N, λ, this extended game, denoted as $\mathsf{Game}^{\mathsf{A}}_{\mathsf{EMH}}$, takes as input a parameter $\bar{k} \in [N+1]$ which is only given to the challenger. $\mathsf{Game}^{\mathsf{A}}_{\mathsf{EMH}}$ proceeds as follows:

Setup. The challenger runs $\mathsf{Setup}_{\mathsf{A}}(\lambda, N)$ and gives the public parameter PP to $\mathcal{A}$.
Phase 1. For $i = 1$ to Q_1, $\mathcal{A}$ adaptively submits (index, attribute set) pair (k_i, S_{k_i}), and obtains $\mathsf{SK}_{k_i,S_{k_i}}$.
Challenge. $\mathcal{A}$ submits two equal-length messages M_0, M_1 and a (revocation list, access policy) pair $(R^*, \mathbb{A}^*)$. The challenger flips a random coin $b \in \{0,1\}$, and sends $CT_{R^*,\mathbb{A}^*} \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M_b, R^*, \mathbb{A}^*, \bar{k})$ to $\mathcal{A}$. This is the only place where $\bar{k}$ is used in the game.
Phase 2. For $i = Q_1 + 1$ to Q, $\mathcal{A}$ adaptively submits (index, attribute set) pair (k_i, S_{k_i}), and obtains $\mathsf{SK}_{k_i,S_{k_i}}$.
Guess. $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ for b.

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

When $\bar{k} = 1$, the game above, including the **restriction**, is exactly identical to the message-hiding game $\mathsf{Game}_{\mathsf{MH}}$ for Σ, we have $\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[1] = \mathsf{MHAdv}_{\mathcal{A}}$. When $\bar{k} = N + 1$, we have that $\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[N+1] \leq \mathsf{MH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}$, since $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}}$ is identical to $\mathsf{Game}^{\mathsf{A}}_{\mathsf{EMH}}$ for $\bar{k} = N + 1$, but there is no restriction in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}}$. In the following proof sketch, we will make use of the facts that Σ_{A} is message-hiding and index-hiding to show that $\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[1]$ is negligible, which implies that $\mathsf{MHAdv}_{\mathcal{A}}$ is negligible (i.e. Σ is secure w.r.t. Def. 1).

Suppose that Σ is not secure, i.e. $\mathsf{MHAdv}_{\mathcal{A}} > \epsilon$ for some adversary $\mathcal{A}$ and non-negligible ϵ. $\mathsf{MHAdv}_{\mathcal{A}} > \epsilon$ implies that $\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[1] > \epsilon$. As Σ_{A} is message-hiding, $\mathsf{MH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}$ is negligible (for simplicity, say $\mathsf{MH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}} = 0$), thus $\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[N+1] = 0$. Then, by the standard hybrid argument there exists a $\bar{k} \in [N]$ such that

$$|\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}] - \mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}+1]| > \epsilon/N.$$

In other words, with non-negligible probability, $\mathcal{A}$ is able to distinguish $\mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, R^*, \mathbb{A}^*, \bar{k})$ from $\mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, R^*, \mathbb{A}^*, \bar{k}+1)$ for some M and $(R^*, \mathbb{A}^*)$. But then $\mathcal{A}$ can directly be used to win the index-hiding game $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$.

More specifically, in Appendix A, we show that for any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that for all $\bar{k} = 1, \ldots, N$, we have

$$|\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}] - \mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}+1]| \leq 2 \cdot \mathsf{IH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{B}}[\bar{k}]. \tag{1}$$

Then we have

$$\begin{aligned} &|\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[1] - \mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[N+1]| \\ \leq &\sum_{\bar{k}=1}^{N} |\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}] - \mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}+1]| \leq 2\sum_{\bar{k}=1}^{N} \mathsf{IH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{B}}[\bar{k}]. \end{aligned}$$

But since Σ_{A} is message-hiding and index-hiding, we have that $\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[N+1]$ and $\mathsf{IH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{B}}[\bar{k}]$ for $\bar{k} = 1, \ldots, N$ are negligible for any PPT adversary. Therefore, $\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[1]$ is negligible. The selective case is similar.

3.2.2 R-CP-ABE Traceability

We now propose a tracing algorithm, which uses a general tracing method previously used in [4,22,6,7,9,20], and show that equipped with Trace, Σ is traceable (w.r.t. Def. 2).

$\mathsf{Trace}^{\mathcal{D}}(\mathsf{PP}, R_{\mathcal{D}}, \mathbb{A}_{\mathcal{D}}, \epsilon) \to \mathbb{K}_T \subseteq [N]$: Given a policy-specific decryption blackbox $\mathcal{D}$ associated with a (revocation list, access policy) pair $(R_{\mathcal{D}}, \mathbb{A}_{\mathcal{D}})$ and probability $\epsilon > 0$, the tracing algorithm works as follows:

1. For $k = 1$ to $N + 1$, do the following:
   (a) Repeat the following $8\lambda(N/\epsilon)^2$ times:
      i. Sample M from the message space at random.
      ii. Let $CT_{R_{\mathcal{D}}, \mathbb{A}_{\mathcal{D}}} \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, R_{\mathcal{D}}, \mathbb{A}_{\mathcal{D}}, k)$.
      iii. Query oracle $\mathcal{D}$ on input $CT_{R_{\mathcal{D}}, \mathbb{A}_{\mathcal{D}}}$, and compare the output of $\mathcal{D}$ with M.
   (b) Let $\hat{p}_k$ be the fraction of times that $\mathcal{D}$ decrypted the ciphertexts correctly.
2. Let $\mathbb{K}_T$ be the set of all $k \in [N]$ for which $\hat{p}_k - \hat{p}_{k+1} \geq \epsilon/(4N)$. Output $\mathbb{K}_T$.

The running time is cubic in N. It can be made (almost) quadratic using binary search instead of a linear scan.

Theorem 2. *If Σ_{A} is message-hiding and index-hiding (resp. selectively index-hiding), then Σ is traceable (resp. selectively traceable).*

Proof. We show that if the blackbox output by the adversary is a useful one then $\mathbb{K}_T$ will satisfy $(\mathbb{K}_T \neq \emptyset) \wedge (\mathbb{K}_T \subseteq \mathbb{K}_{\mathcal{D}}) \wedge (\exists k_t \in \mathbb{K}_T \; s.t. \; (k_t \in [N] \setminus R_{\mathcal{D}}) \wedge (S_{k_t} \; satisfies \; \mathbb{A}_{\mathcal{D}}))$ with overwhelming probability, which implies that the adversary cannot win $\mathsf{Game}_{\mathsf{TR}}$, i.e., $\mathsf{TRAdv}_{\mathcal{A}}$ is negligible. The selective case will be similar. Let $\mathcal{D}$ be the policy-specific decryption blackbox output by the adversary, and $(R_{\mathcal{D}}, \mathbb{A}_{\mathcal{D}})$ be the (revocation list, access policy) pair describing $\mathcal{D}$. Define

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

where the probability is taken over the random choice of message M and the random coins of $\mathcal{D}$. We have that $p_1 \geq \epsilon$ and p_{N+1} is negligible (for simplicity let $p_{N+1} = 0$). The former follows from the fact that $\mathcal{D}$ is useful, and the latter is because Σ_{A} is message-hiding in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}}$. Then there must exist some $k \in [N]$ such that $p_k - p_{k+1} \geq \epsilon/(2N)$. By the Chernoff bound it follows that with overwhelming probability, $\hat{p}_k - \hat{p}_{k+1} \geq \epsilon/(4N)$. Hence, we have $\mathbb{K}_T \neq \emptyset$.

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

4 An Efficient Augmented R-CP-ABE

We propose an AugR-CP-ABE scheme which is highly expressive and efficient with sub-linear overhead in the number of users in the system. It is also *large universe*, where attributes do not need to be enumerated during setup, and the public parameter size is independent of the attribute universe size. We show that this AugR-CP-ABE is message-hiding and selectively index-hiding in the standard model.

Combining this AugR-CP-ABE with the results in Sec. 3.2, we obtain a large universe R-CP-ABE which is selectively secure and traceable, and for a fully collusion-resistant blackbox traceable system, the resulting R-CP-ABE is the most efficient one to date, with sub-linear overhead.

To obtain this practical CP-ABE scheme supporting traitor tracing, revocation and large universe, we borrow ideas from the Blackbox Traceable CP-ABE of [20], the Trace and Revoke scheme of [9] and the Large Universe CP-ABE of [26], but the work is not trivial as a straightforward combination of the ideas would result in a scheme which is inefficient, insecure, or is not able to achieve strong traceability, as also discussed in [20]. Specifically, by incorporating the ideas from [9] and [26] into the Augmented CP-ABE of [20], we can obtain a large universe AugR-CP-ABE which is message-hiding, but proving the index-hiding property is a challenging task. The proof techniques for index-hiding in [20] only work if the attribute universe size is polynomial in the security parameter and the parameters of attributes have to be enumerated during setup. They are not applicable to large universe. The proof techniques in [26] are applicable to large universe, but work only for proving security (i.e. message-hiding), while not applicable to index-hiding. To prove index-hiding in the large universe setting, we introduce a new assumption that the index-hiding of our large universe AugR-CP-ABE can be based on. In particular, in the underlying q-1 assumption of [26] on bilinear groups $(p, \mathbb{G}, \mathbb{G}_T, e)$, the challenge term $T \in \mathbb{G}_T$ is $e(g,g)^{ca^{q+1}}$ or a random element, and such a term in the target group could be used to prove the message-hiding as the message space is

$\mathbb{G}_T$. To prove the index-hiding, which is based on the ciphertext components in the source group $\mathbb{G}$, we need the challenge term to be in the source group $\mathbb{G}$. Inspired by the Source Group q-Parallel BDHE Assumption in [19], which is a close relative to the (target group) Decisional Parallel BDHE Assumption in [30], we modify the q-1 assumption to its source group version where the challenge term is $g^{ca^{q+1}}$ or a random element in $\mathbb{G}$. Based on this new assumption and with a new crucial proof idea, we prove the index-hiding property for our large universe AugR-CP-ABE. We prove that this new assumption holds in the generic group model.

4.1 Preliminaries

Linear Secret-Sharing Schemes (LSSS). An LSSS is a share-generating matrix A whose rows labeled by attributes via a function ρ. An attribute set S satisfies the LSSS access matrix A if the rows labeled by the attributes in S have the *linear reconstruction* property, namely, there exist constants $\{\omega_i\}$ such that, for any valid shares $\{\lambda_i\}$ of a secret s, we have $\sum_i \omega_i \lambda_i = s$. The formal definitions of access structures and LSSS can be found in [18,26].

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. Besides the Decision 3-Party Diffie-Hellman Assumption (D3DH) and the Decisional Linear Assumption (DLIN) that are used in [9] to achieve traceability in broadcast encryption, the index-hiding property of our AugR-CP-ABE construction will rely on a new assumption, which is similar to the Source Group q-Parallel BDHE Assumption [19]. We refer to it as the Modified Source Group q-Parallel BDHE Assumption. Here we only review this new assumption, and refer to [9] for the details of the other assumptions.

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

$$
\begin{aligned}
&(p, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}, \quad g \xleftarrow{R} \mathbb{G}, \ a, c, d, b_1, \ldots, b_q \xleftarrow{R} \mathbb{Z}_p, \\
&D = \big((p, \mathbb{G}, \mathbb{G}_T, e), \ g, g^d, g^{cd}, g^{da^q}, \\
&\qquad g^{a^i}, \ g^{b_j}, \ g^{a^i b_j}, \ g^{a^i/b_j^2}, \ g^{cdb_j} \quad \forall i, j \in [q], \\
&\qquad g^{a^i/b_j} \qquad\qquad \forall i \in [2q] \setminus \{q+1\}, j \in [q], \\
&\qquad g^{a^i b_{j'}/b_j^2} \qquad\qquad \forall i \in [2q], j, j' \in [q] \ s.t. \ j' \neq j, \\
&\qquad g^{cda^i b_{j'}/b_j}, g^{cda^i b_{j'}/b_j^2} \qquad \forall i \in [q], j, j' \in [q] \ s.t. \ j \neq j' \ \big), \\
&T_0 = g^{ca^{q+1}}, T_1 \xleftarrow{R} \mathbb{G}.
\end{aligned}
$$

The advantage of an algorithm $\mathcal{A}$ in breaking the Modified Source Group q-Parallel BDHE Assumption is: $Adv^q_{\mathcal{G},\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(D, T_0) = 1] - \Pr[\mathcal{A}(D, T_1) = 1]|$.

Definition 5. *$\mathcal{G}$ satisfies the Modified Source Group q-Parallel BDHE Assumption if $Adv^q_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ for any PPT algorithm $\mathcal{A}$.*

This new assumption is closely related to the q-1 assumption in [26], except that the challenge term $g^{ca^{q+1}}$ remains in the source group, all the input terms replace c with cd, and additional input terms g^d and g^{da^q} are given to the adversary. The relation between this assumption and the q-1 assumption [26] is analogous to that between the Source Group q-Parallel BDHE Assumption [19] and the Decisional Parallel BDHE Assumption [30], i.e. the challenge term changes from a term in the target group (i.e. $e(g,g)^{ca^{q+1}}$) to a term in the source group (i.e. $g^{ca^{q+1}}$), and the input terms are modified accordingly (i.e. replacing c with cd, and adding g^d). The main difference is that in this new assumption, there is an additional input term g^{da^q}. Note that giving the term g^{da^q} does not pose any problem in the generic group model. Intuitively, there are two ways that the adversary may make use of the term g^{da^q}: (1) pairing g^{da^q} with the challenge term: since the pairing result of any two input terms would not be $e(g,g)^{cda^{2q+1}}$, the adversary cannot break this new assumption in this way; (2) pairing the challenge term with another input term whose exponent contains d: however, the result could be a random element or one of $\{\ e(g,g)^{cda^{q+1}},\ e(g,g)^{c^2da^{q+1}},\ e(g,g)^{c^2db_ja^{q+1}},\ e(g,g)^{c^2da^{q+1+i}b_{j'}/b_j},\ e(g,g)^{c^2da^{q+1+i}b_{j'}/b_j^2}\}$, and as there is no input term which can be paired with g^{da^q} to obtain any of these terms, the adversary cannot break this new assumption by this way either. In Appendix D, we prove that this assumption holds in the generic group model.

Notations. Suppose that the number of users N in the system equals to m^2 for some m. In practice, if N is not a square, we can add some "dummy" users until it pads to the next square. We arrange the users in an $m \times m$ matrix and uniquely assign a tuple (i,j), where $i,j \in [m]$, to each user. A user at position (i,j) of the matrix has index $k = (i-1) * m + j$. For simplicity, we directly use (i,j) as the index where $(i,j) \geq (\bar{i},\bar{j})$ means that $((i > \bar{i}) \vee (i = \bar{i} \wedge j \geq \bar{j}))$. Let $[m,m]$ be the set $\{(i,j)|i,j \in [m]\}$. The use of pairwise notation (i,j) is purely a notational convenience, as $k = (i-1) * m + j$ defines a bijection between $\{(i,j)|i,j \in [m]\}$ and $[N]$. For a given vector $\boldsymbol{v} = (v_1,\ldots,v_d)$, by $g^{\boldsymbol{v}}$ we mean the vector $(g^{v_1},\ldots,g^{v_d})$. Furthermore, for $g^{\boldsymbol{v}} = (g^{v_1},\ldots,g^{v_d})$ and $g^{\boldsymbol{w}} = (g^{w_1},\ldots,g^{w_d})$, by $g^{\boldsymbol{v}} \cdot g^{\boldsymbol{w}}$ we mean the vector $(g^{v_1+w_1},\ldots,g^{v_d+w_d})$, i.e. $g^{\boldsymbol{v}} \cdot g^{\boldsymbol{w}} = g^{\boldsymbol{v}+\boldsymbol{w}}$, and by $e_d(g^{\boldsymbol{v}}, g^{\boldsymbol{w}})$ we mean $\prod_{k=1}^{d} e(g^{v_k}, g^{w_k})$, i.e. $e_d(g^{\boldsymbol{v}}, g^{\boldsymbol{w}}) = e(g,g)^{(\boldsymbol{v}\cdot\boldsymbol{w})}$, where $(\boldsymbol{v}\cdot\boldsymbol{w})$ is the inner product of $\boldsymbol{v}$ and $\boldsymbol{w}$. Given a bilinear group order 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 $\mathbb{Z}_p^3 = span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2, \boldsymbol{\chi}_3\} = \{\nu_1\boldsymbol{\chi}_1 + \nu_2\boldsymbol{\chi}_2 + \nu_3\boldsymbol{\chi}_3 | \nu_1, \nu_2, \nu_3 \in \mathbb{Z}_p\}$. For any $\boldsymbol{v} \in span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$, $(\boldsymbol{\chi}_3 \cdot \boldsymbol{v}) = 0$, and for random $\boldsymbol{v} \in \mathbb{Z}_p^3$, $(\boldsymbol{\chi}_3 \cdot \boldsymbol{v}) \neq 0$ happens with overwhelming probability.

4.2 Augmented R-CP-ABE Construction

Now we propose a large universe Augmented R-CP-ABE, where the attribute universe is $\mathcal{U} = \mathbb{Z}_p$, and we do not need to enumerate all the attributes or their corresponding public parameters during system setup.

$\mathsf{Setup}_{\mathsf{A}}(\lambda, N = m^2) \rightarrow (\mathsf{PP}, \mathsf{MSK})$. The algorithm calls the group generator $\mathcal{G}(1^\lambda)$ to get $(p, \mathbb{G}, \mathbb{G}_T, e)$, where p is the prime order of $\mathbb{G}$ and $\mathbb{G}_T$ and e is the bilinear map, and sets the attribute universe to $\mathcal{U} = \mathbb{Z}_p$. It then randomly picks:

$$g,\ h,\ f,\ f_1,\ldots,f_m,\ G,\ H\ \in \mathbb{G},\quad \{\alpha_i,\ r_i,\ z_i \in \mathbb{Z}_p\}_{i\in[m]},\ \{c_j \in \mathbb{Z}_p\}_{j\in[m]},$$

and outputs the public parameter PP and master secret key MSK as

$$\mathsf{PP} = \Big(\ (p, \mathbb{G}, \mathbb{G}_T, e),\ g,\ h,\ f,\ f_1,\ldots,f_m,\ G,\ H,$$

$$\{E_i = e(g,g)^{\alpha_i}, G_i = g^{r_i},\ Z_i = g^{z_i}\}_{i\in[m]},\quad \{H_j = g^{c_j}\}_{j\in[m]}\ \Big).$$

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

A counter $ctr = 0$ is implicitly included in MSK.

$\mathsf{KeyGen}_\mathsf{A}(\mathsf{PP}, \mathsf{MSK}, S \subseteq \mathbb{Z}_p) \to \mathsf{SK}_{(i,j),S}$. The algorithm first sets $ctr = ctr + 1$ and computes the corresponding index in the form of (i,j) where $1 \le i,j \le m$ and $(i-1) * m + j = ctr$. Then it picks random exponents $\sigma_{i,j} \in \mathbb{Z}_p$, $\{\delta_{i,j,x} \in \mathbb{Z}_p\}_{\forall x\in S}$, and outputs a secret key $\mathsf{SK}_{(i,j),S} = \Big((i,j), S,\ K_{i,j}, K'_{i,j}, K''_{i,j},\ \{\bar{K}_{i,j,j'}\}_{j'\in[m]\setminus\{j\}},\ \{K_{i,j,x}, K'_{i,j,x}\}_{x\in S}\Big)$ where

$$K_{i,j} = g^{\alpha_i} g^{r_i c_j} (f f_j)^{\sigma_{i,j}},\ K'_{i,j} = g^{\sigma_{i,j}},\ K''_{i,j} = Z_i^{\sigma_{i,j}},\quad \{\bar{K}_{i,j,j'} = f_{j'}^{\sigma_{i,j}}\}_{j'\in[m]\setminus\{j\}},$$
$$\{K_{i,j,x} = g^{\delta_{i,j,x}},\ K'_{i,j,x} = (H^x h)^{\delta_{i,j,x}} G^{-\sigma_{i,j}}\}_{x\in S}.$$

$\mathsf{Encrypt}_\mathsf{A}(\mathsf{PP}, M, R, \mathbb{A} = (A,\rho), (\bar{i},\bar{j})) \to CT_{R,(A,\rho)}$. $R \subseteq [m,m]$ is a revocation list. $\mathbb{A} = (A,\rho)$ is an LSSS matrix where A is an $l \times n$ matrix and ρ maps each row A_k of A to an attribute $\rho(k) \in \mathcal{U} = \mathbb{Z}_p$. This algorithm allows the encrypting party to encrypt a message to the recipients whose (index, attribute set) pairs $((i,j), S_{(i,j)})$ satisfy $\big((i,j) \in [m,m] \setminus R\big) \wedge \big(S_{(i,j)}\ \ satisfies\ (A,\rho)\big) \wedge \big((i,j) \ge (\bar{i},\bar{j})\big)$. Let $\bar{R} = [m,m] \setminus R$ and for $i \in [m]$, $\bar{R}_i = \{j' | (i,j') \in \bar{R}\}$, that is, $\bar{R}$ is the non-revoked index list, and $\bar{R}_i$ is the set of non-revoked column index on the i-th row. The algorithm randomly chooses

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

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

$$\boldsymbol{v}_i \in \mathbb{Z}_p^3\ \forall i \in \{1,\ldots,\bar{i}\},$$
$$\boldsymbol{v}_i \in span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}\ \forall i \in \{\bar{i}+1,\ldots,m\},$$

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

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

$$\boldsymbol{R}_i = g^{\boldsymbol{v}_i},\ \boldsymbol{R}'_i = g^{\kappa \boldsymbol{v}_i},\quad Q_i = g^{s_i},\ Q'_i = (f \prod_{j'\in\bar{R}_i} f_{j'})^{s_i} Z_i^{t_i} f^{\pi},\ Q''_i = g^{t_i},\quad T_i = E_i^{\hat{s}_i}.$$

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

$$\boldsymbol{R}_i = G_i^{s_i \boldsymbol{v}_i},\quad \boldsymbol{R}'_i = G_i^{\kappa s_i \boldsymbol{v}_i},$$
$$Q_i = g^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)},\quad Q'_i = (f \prod_{j'\in\bar{R}_i} f_{j'})^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)} Z_i^{t_i} f^{\pi},\quad Q''_i = g^{t_i},\quad T_i = M \cdot E_i^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}.$$

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

$\mathsf{Decrypt}_{\mathsf{A}}(\mathsf{PP}, CT_{R,(A,\rho)}, \mathsf{SK}_{(i,j),S}) \rightarrow M$ or $\perp$. For ciphertext $CT_{R,(A,\rho)} = \langle R, (A,\rho), (\boldsymbol{R}_i, \boldsymbol{R}'_i, Q_i, Q'_i, Q''_i, T_i)_{i=1}^m, (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^m, (P_k, P'_k, P''_k)_{k=1}^l \rangle$ and secret key $\mathsf{SK}_{(i,j),S} = \Big((i,j), S, K_{i,j}, K'_{i,j}, K''_{i,j}, \{\bar{K}_{i,j,j'}\}_{j' \in [m] \setminus \{j\}}, \{K_{i,j,x}, K'_{i,j,x}\}_{x \in S}\Big)$, if $(i,j) \in R$ or S does not satisfy (A,ρ), the algorithm outputs $\perp$, otherwise:
1. Since S satisfies (A,ρ), the algorithm can efficiently compute constants $\{\omega_k \in \mathbb{Z}_p\}$ such that $\sum_{\rho(k) \in S} \omega_k A_k = (1, 0, \ldots, 0)$, then compute

$$\begin{aligned} D_P &= \prod_{\rho(k) \in S} \big(e(K'_{i,j}, P_k) \cdot e(K_{i,j,\rho(k)}, P'_k) \cdot e(K'_{i,j,\rho(k)}, P''_k)\big)^{\omega_k} \\ &= \prod_{\rho(k) \in S} \big(e(g^{\sigma_{i,j}}, f^{A_k \cdot \boldsymbol{u}} G^{\xi_k}) \cdot e(g^{\delta_{i,j,\rho(k)}}, (H^{\rho(k)} h)^{-\xi_k}) \cdot e((H^{\rho(k)} h)^{\delta_{i,j,\rho(k)}} G^{-\sigma_{i,j}}, g^{\xi_k})\big)^{\omega_k} \\ &= \prod_{\rho(k) \in S} \big(e(g^{\sigma_{i,j}}, f^{A_k \cdot \boldsymbol{u}})\big)^{\omega_k} = e(g^{\sigma_{i,j}}, f)^{\pi}. \end{aligned}$$

Note that if S does not satisfy (A,ρ), no such constants $\{\omega_k \in \mathbb{Z}_p\}$ would exist.
2. Since $(i,j) \in \bar{R} (= [m,m] \setminus R)$ implies $j \in \bar{R}_i$, the algorithm can compute

$$\bar{K}_{i,j} = K_{i,j} \cdot \Big(\prod_{j' \in \bar{R}_i \setminus \{j\}} \bar{K}_{i,j,j'}\Big) = g^{\alpha_i} g^{r_i c_j} (f f_j)^{\sigma_{i,j}} \cdot \Big(\prod_{j' \in \bar{R}_i \setminus \{j\}} f_{j'}^{\sigma_{i,j}}\Big) = g^{\alpha_i} g^{r_i c_j} \cdot \Big(f \prod_{j' \in \bar{R}_i} f_{j'}\Big)^{\sigma_{i,j}}.$$

Note that if $(i,j) \in R$ (implying $j \notin \bar{R}_i$), the algorithm cannot produce such a $\bar{K}_{i,j}$. The algorithm then computes

$$D_I = \frac{e(\bar{K}_{i,j}, Q_i) \cdot e(K''_{i,j}, Q''_i)}{e(K'_{i,j}, Q'_i)} \cdot \frac{e_3(\boldsymbol{R}'_i, \boldsymbol{C}'_j)}{e_3(\boldsymbol{R}_i, \boldsymbol{C}_j)}.$$

3. Computes $M = T_i / (D_P \cdot D_I)$. Suppose that the ciphertext is generated from message M' and encryption index $(\bar{i}, \bar{j})$, it can be verified that only when $(i > \bar{i})$ or $(i = \bar{i} \wedge j \geq \bar{j})$, $M = M'$. This is because 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 is given in Appendix B.

4.3 Augmented R-CP-ABE Security

The following theorem states that the AugR-CP-ABE proposed above is message-hiding. Then in Theorem 4, we state that the AugR-CP-ABE is also selectively index-hiding.

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

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

Theorem 4. *Suppose that the D3DH, the DLIN and the Modified Source Group q-Parallel BDHE Assumption hold. Then no PPT adversary can selectively win* $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$ *with non-negligible advantage, provided that the challenge LSSS matrix's size* $l \times n$ *satisfies* $l, n \leq q$.

Proof. It follows Lemma 1 and Lemma 2 below.

Lemma 1. *If the D3DH and the Modified Source Group q-Parallel BDHE Assumption hold, then for* $\bar{j} < m$, *no PPT adversary can selectively distinguish between an encryption to* $(\bar{i}, \bar{j})$ *and* $(\bar{i}, \bar{j}+1)$ *in* $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$ *with non-negligible advantage, provided that the challenge LSSS matrix's size* $l \times n$ *satisfies* $l, n \leq q$.

Proof. In $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$ with index $(\bar{i}, \bar{j})$, let $(R^*, (A^*, \rho^*))$ be the challenge (revocation list, access policy) pair, the restriction is that the adversary $\mathcal{A}$ does not query a decryption key for (index, attribute set) pair $((i,j), S_{(i,j)})$ such that $\big((i,j) = (\bar{i}, \bar{j})\big) \wedge \big((i,j) \in [m,m] \setminus R^*\big) \wedge \big(S_{(i,j)}\ satisfies\ (A^*, \rho^*)\big)$. Under this restriction, there are two ways for $\mathcal{A}$ to take:

Case I: In Phase 1 and Phase 2, $\mathcal{A}$ does not query a decryption key with index $(\bar{i}, \bar{j})$.
Case II: In Phase 1 or Phase 2, $\mathcal{A}$ queries a decryption key with index $(\bar{i}, \bar{j})$. Let $S_{(\bar{i},\bar{j})}$ be the corresponding attribute set. **Case II** has the following sub-cases:
1. $(\bar{i}, \bar{j}) \notin [m,m] \setminus R^*$, $S_{(\bar{i},\bar{j})}$ satisfies (A^*, ρ^*).
2. $(\bar{i}, \bar{j}) \notin [m,m] \setminus R^*$, $S_{(\bar{i},\bar{j})}$ does not satisfy (A^*, ρ^*).
3. $(\bar{i}, \bar{j}) \in [m,m] \setminus R^*$, $S_{(\bar{i},\bar{j})}$ does not satisfy (A^*, ρ^*).

If $\mathcal{A}$ is in **Case I**, **Case II.1** or **Case II.2**, it follows the restrictions in the index-hiding game for Augmented Broadcast Encryption (AugBE) in [9], where the adversary does not query the key with index $(\bar{i}, \bar{j})$ or $(\bar{i}, \bar{j})$ is not in the receiver list $[m,m] \setminus R^*$. **Case II.3** captures the index-hiding requirement of Augmented R-CP-ABE in that even if a user has a key with index $(\bar{i}, \bar{j})$ and $(\bar{i}, \bar{j}) \notin R^*$, the user cannot distinguish between an encryption to $(R^*, (A^*, \rho^*), (\bar{i}, \bar{j}))$ and $(R^*, (A^*, \rho^*), (\bar{i}, \bar{j}+1))$ if the corresponding attribute set $S_{(\bar{i},\bar{j})}$ does not satisfy (A^*, ρ^*). This is the most challenging part of proving the index-hiding when we attempt to *securely intertwine* the tracing techniques of broadcast encryption (e.g. [9]) into the large universe CP-ABE (e.g. [26]). Compared to the proof of [20], the challenge here is to prove the index-hiding in the large universe setting, as discussed previously.

To prove this lemma, we flip a random coin $c \in \{0,1\}$ as our guess on which case that $\mathcal{A}$ is in. If $\mathcal{A}$ is in **Case I**, **Case II.1** or **Case II.2**, we make a reduction that uses $\mathcal{A}$ to solve a D3DH problem instance, using a proof technique similar to that of [9]. Actually, in this proof, we reduce from our AugR-CP-ABE to the AugBE in [9]. If $\mathcal{A}$ is in **Case I**, **Case II.2** or **Case II.3**, we use $\mathcal{A}$ to solve a Modified Source Group q-Parallel BDHE problem instance, which is where the main novelty resides among all the proofs in this work. Please refer to Appendix C for details.

Lemma 2. *If the D3DH, the DLIN and the Modified Source Group q-Parallel BDHE Assumption hold, then for* $1 \leq \bar{i} \leq m$, *no PPT adversary can selectively distinguish between an encryption to* $(\bar{i}, m)$ *and* $(\bar{i}+1, 1)$ *in* $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$ *with non-negligible advantage, provided that the challenge LSSS matrix's size* $l \times n$ *satisfies* $l, n \leq q$.

Proof. Similar to the proof of Lemma 6.3 in [9], to prove this lemma we define the following hybrid experiment: H_1: encrypt to $(\bar{i}, \bar{j} = m)$; H_2: encrypt to $(\bar{i}, \bar{j} = m+1)$; and H_3: encrypt to $(\bar{i}+1, 1)$. This lemma follows Claim 1 and Claim 2 below.

Claim 1. *If the D3DH and the Modified Source Group q-Parallel BDHE Assumption hold, then no PPT adversary can selectively distinguish between experiment H_1 and H_2 with non-negligible advantage, provided that the challenge LSSS matrix's size $l \times n$ satisfies $l, n \leq q$.*

Proof. The proof is identical to that for Lemma 1.

Claim 2. *If the D3DH and the DLIN hold, then no PPT adversary can distinguish between experiment H_2 and H_3 with non-negligible advantage.*

Proof. Note that $(\bar{i}, m+1) \notin [m, m]$ implies that for any revocation list $R^* \subseteq [m, m]$, we have $(\bar{i}, m+1) \notin \bar{R}^* (= [m, m] \setminus R^*)$, i.e, the adversaries for distinguishing H_2 and H_3 will not be in **Case II.3**. Thus, we can prove this claim in a similar way to that of [9]. Actually, in this proof, we reduce from our AugR-CP-ABE to the AugBE in [9]. In the proof of index-hiding for an AugBE scheme Σ_{AugBE} in [9, Lemma 6.3], two hybrid experiments were defined and proven indistinguishable via a sequence of hybrid sub-experiments.

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

By following [9, Lemma 6.3], *if the D3DH and the DLIN hold, no PPT adversary can distinguish between H_2^{AugBE} and H_3^{AugBE} with non-negligible advantage for Σ_{AugBE}.* Suppose there is a PPT adversary $\mathcal{A}$ that can distinguish between H_2 and H_3 for Σ_{A} with non-negligible advantage. We can build a reduction, which is similar to that of **Case A** in Appendix C, to use $\mathcal{A}$ to distinguish between H_2^{AugBE} and H_3^{AugBE} for Σ_{AugBE} with non-negligible advantage.

5 KP-ABE Analog

We have obtained the first practical CP-ABE scheme that simultaneously supports (1) public and fully collusion-resistant traceability against policy-specific decryption blackbox, (2) (direct) revocation and (3) "truly" large attribute universe, and is also highly expressive (i.e. supporting any monotonic access structures) and efficient (i.e. enjoying the sub-linear overhead of $O(\sqrt{N})$ while supporting fully collusion-resistant blackbox traceability). The scheme's security and traceability are proven against selectively adversaries in the standard model. Our techniques also yield an analogous Key-Policy ABE (KP-ABE) scheme, i.e. the first practical KP-ABE scheme that simultaneously supports (1) public and fully collusion-resistant traceability against attributes-specific decryption blackbox, (2) (direct) revocation and (3) "truly" large attribute universe, and is also highly expressive (i.e. supporting any monotonic access structures) and efficient (i.e. enjoying the sub-linear overhead of $O(\sqrt{N})$ while supporting fully collusion-resistant blackbox traceability). Essentially, KP-ABE is like CP-ABE with the roles of keys and ciphertexts reversed: in KP-ABE, keys are associated with access policies and ciphertexts are associated with sets of attributes. In the setting of KP-ABE, attributes-specific decryption blackbox, which can decrypt ciphertxets generated under some specific attribute set, reflects more general and practical applications than key-like decryption blackbox which functions like a private key with certain access policy. Our techniques readily adapt to KP-ABE and attributes-specific decryption blackbox, and the definitions, constructions and proofs are very similar to the CP-ABE case. The details can be found in Appendix E, Appendix F, and Appendix G. In Appendix E we present the definition for KP-ABE supporting traceability against attributes-specific decryption blackbox, direct revocation and large attribute

[1] [2]	Blackbox Traceability	Revocation	Large Universe	Public Key Size	Ciphertext Size	Private Key Size	Pairings in Decryption
[1, Sec. 4.1]	×	√	∂ [3]	$2N+2+m$ [3]	$3+\|S\|$	$2l$	$2+2\|I\|$
[1, Sec. 4.2]	×	√	∂ [3]	$6+m$ [3]	$2+\|S\|+2\|R\|$ [3]	$2+2l$	$2\|I\|+2\|R\|$ [3]
[26]	×	×	√	5	$2+2\|S\|$	$3l$	$3\|I\|$
this work	√	√	√	$5+5\sqrt{N}$	$1+16\sqrt{N}+2\|S\|$	$2+\sqrt{N}+3l$	$9+3\|I\|$

[1] All the four schemes are highly expressive, i.e. supporting any monotonic access structures.

[2] Let N be the number of users in the system, l the number of rows of the LSSS matrix for an access policy, $|S|$ the size of the attribute set of a ciphertext, and $|I|$ the number of attributes for a ciphertext to satisfy a key policy.

[3] The KP-ABE schemes in [1] are not "truly" large universe, as some limitations are imposed and some corresponding parameters have to be fixed during the setup. Let m be the maximum size of an attribute set associated with a ciphertext, and $|R|$ the number of revoked users in a revocation list R.

Table 2. KP-ABE: Features and Efficiency Comparison

universe, and call it Revocable KP-ABE (R-KP-ABE). In Appendix F we extend the R-KP-ABE to a primitive called Augmented R-KP-ABE (or AugR-KP-ABE for short), then formalize a message-hiding game and an index-hiding game, and show that an AugR-KP-ABE scheme satisfying the message-hiding and the (selective) index-hiding can be transformed to a (selectively) secure R-KP-ABE scheme with (selective) attributes-specific blackbox traceability. In Appendix G we propose a *large universe* AugR-KP-ABE, and prove that it is message-hiding and selective index-hiding in the standard model. Combining it with the results in Appendix F, we obtain a large universe R-KP-ABE construction, which is efficient (with overhead size in $O(\sqrt{N})$), highly expressive (supporting any monotonic access structures as policies), selectively secure and selectively attributes-specific blackbox traceable in the standard model.

Table 2 compares this new KP-ABE scheme with the representative results in revocable KP-ABE [1] and "truly" large universe KP-ABE [26], in terms of features (i.e. blackbox traceability, revocation and large universe) and performance. The scheme's overhead is in $O(\sqrt{N})$, where N is the number of users in a system. This might be a concern, but we stress that for fully collusion-resistant blackbox traceable KP-ABE, such a sub-linear overhead is the most efficient one to date. It is worth mentioning that the traceable Predicate Encryption (PE) scheme by Katz and Schröder [14] implies an expressive KP-ABE scheme with fully collusion-resistant blackbox traceability, but the scheme's overhead is linear in N, and it does not support revocation or "truly" large universe.

6 Conclusion

In this paper, we proposed the first practical CP-ABE and KP-ABE that simultaneously support (1) traitor tracing, (2) revocation and (3) large universe. Both schemes are highly expressive in supporting any monotonic access structures. Besides achieving fully collusion-resistant blackbox traceability, and direct revocation, they are also efficient with the overhead in $O(\sqrt{N})$ only. Furthermore, they support large attribute universe and do not need to fix the values of attributes during the system setup. The CP-ABE (resp. KP-ABE) scheme was proven selectively secure and selectively traceable against policy-specific (resp. attributes-specific) decryption blackbox in the standard model.

References

1. Attrapadung, N., Imai, H.: Conjunctive broadcast and attribute-based encryption. In: Pairing. pp. 248–265 (2009)
2. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy. pp. 321–334. IEEE Computer Society (2007)
3. Boneh, D., Boyen, X., Goh, E.J.: Hierarchical identity based encryption with constant size ciphertext. In: EUROCRYPT. pp. 440–456 (2005)
4. Boneh, D., Franklin, M.K.: An efficient public key traitor tracing scheme. In: CRYPTO. pp. 338–353 (1999)
5. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. In: CRYPTO. pp. 213–229 (2001)
6. Boneh, D., Sahai, A., Waters, B.: Fully collusion resistant traitor tracing with short ciphertexts and private keys. In: EUROCRYPT. pp. 573–592 (2006)
7. Boneh, D., Waters, B.: A fully collusion resistant broadcast, trace, and revoke system. In: ACM Conference on Computer and Communications Security. pp. 211–220 (2006)
8. Cheung, L., Newport, C.C.: Provably secure ciphertext policy ABE. In: ACM Conference on Computer and Communications Security. pp. 456–465 (2007)
9. Garg, S., Kumarasubramanian, A., Sahai, A., Waters, B.: Building efficient fully collusion-resilient traitor tracing and revocation schemes. In: ACM Conference on Computer and Communications Security. pp. 121–130 (2010)
10. Goyal, V., Jain, A., Pandey, O., Sahai, A.: Bounded ciphertext policy attribute based encryption. In: ICALP (2). pp. 579–591 (2008)
11. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM Conference on Computer and Communications Security. pp. 89–98 (2006)
12. Herranz, J., Laguillaumie, F., Ràfols, C.: Constant size ciphertexts in threshold attribute-based encryption. In: Public Key Cryptography. pp. 19–34 (2010)
13. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: EUROCRYPT. pp. 146–162 (2008)
14. Katz, J., Schröder, D.: Tracing insider attacks in the context of predicate encryption schemes. In: ACITA (2011), `https://www.usukita.org/node/1779`
15. Lewko, A.B.: Tools for simulating features of composite order bilinear groups in the prime order setting. In: EUROCRYPT. pp. 318–335 (2012)
16. Lewko, A.B., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In: EUROCRYPT. pp. 62–91 (2010)
17. Lewko, A.B., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. IACR Cryptology ePrint Archive 2010, 110 (2010)
18. Lewko, A.B., Waters, B.: New proof methods for attribute-based encryption: Achieving full security through selective techniques. In: CRYPTO. pp. 180–198 (2012)
19. 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)
20. 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. ACM (2013)
21. Liu, Z., Cao, Z., Wong, D.S.: White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures. IEEE Transactions on Information Forensics and Security 8(1), 76–88 (2013)
22. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: CRYPTO. pp. 41–62 (2001)
23. Naor, M., Pinkas, B.: Efficient trace and revoke schemes. In: Financial Cryptography. pp. 1–20 (2000)
24. Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: CRYPTO. pp. 191–208 (2010)
25. Okamoto, T., Takashima, K.: Fully secure unbounded inner-product and attribute-based encryption. In: ASIACRYPT. pp. 349–366 (2012)
26. 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)
27. Sahai, A., Seyalioglu, H., Waters, B.: Dynamic credentials and ciphertext delegation for attribute-based encryption. In: CRYPTO. pp. 199–217 (2012)
28. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: EUROCRYPT. pp. 457–473 (2005)
29. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: EUROCRYPT. pp. 256–266 (1997)
30. Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: Public Key Cryptography. pp. 53–70 (2011)
31. Yu, S., Wang, C., Ren, K., Lou, W.: Attribute based data sharing with attribute revocation. In: ASIACCS. pp. 261–270 (2010)

A AugR-CP-ABE Implies Secure R-CP-ABE

To prove that the R-CP-ABE scheme Σ in Sec. 3.2 is secure it remains to prove that Equation (1) holds for all $\bar{k} = 1, \ldots, N$. Consider a specific $\bar{k} \in [N]$. Adversary $\mathcal{B}$ plays the index-hiding game $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$ with input $\bar{k}$ and works as follows:

Setup. $\mathcal{B}$ receives PP from its challenger in the index-hiding game $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$. $\mathcal{B}$ runs adversary $\mathcal{A}$ in the extend message-hiding game $\mathsf{Game}^{\mathsf{A}}_{\mathsf{EMH}}$ and gives PP to $\mathcal{A}$.

Phase 1. For $i = 1$ to Q_1, $\mathcal{A}$ adaptively submits (index, attribute set) pair (k_i, S_{k_i}) to $\mathcal{B}$. $\mathcal{B}$ submits (k_i, S_{k_i}) to the challenger and receives secret key $\mathsf{SK}_{k_i,S_{k_i}}$. Then $\mathcal{B}$ gives $\mathsf{SK}_{k_i,S_{k_i}}$ to $\mathcal{A}$.

Challenge. $\mathcal{A}$ submits two equal-length messages M_0, M_1 and a (revocation list, access policy) $(R^*, \mathbb{A}^*)$ to $\mathcal{B}$, under the restriction that none of the queried pairs $\{(k_i, S_{k_i})\}_{i=1}^{Q_1}$ can satisfy $(k_i \in [N] \setminus R^*) \wedge (S_{k_i}\ satisfies\ \mathbb{A}^*)$. $\mathcal{B}$ flips a coin $\gamma \in \{0,1\}$, then gives M_γ and $(R^*, \mathbb{A}^*)$ to its challenger. Note that $(R^*, \mathbb{A}^*)$ satisfies the restriction on $\mathcal{B}$ in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$ that none of the queried pairs $\{(k_i, S_{k_i})\}_{i=1}^{Q_1}$ can satisfy $(k_i = \bar{k}) \wedge (k_i \in [N] \setminus R^*) \wedge (S_{k_i}\ satisfies\ \mathbb{A}^*)$. $\mathcal{B}$ receives $CT_{R^*,\mathbb{A}^*} \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M_\gamma, R^*, \mathbb{A}^*, \bar{k} + b)$ for some random $b \in \{0,1\}$. Then $\mathcal{B}$ gives $CT_{R^*,\mathbb{A}^*}$ to $\mathcal{A}$.

Phase 2. For $i = Q_1{+}1$ to Q, $\mathcal{A}$ adaptively submits (index, attribute set) pair (k_i, S_{k_i}) to $\mathcal{B}$, under the restriction that (k_i, S_{k_i}) does not satisfy $(k_i \in [N] \setminus R^*) \wedge (S_{k_i}\ satisfies\ \mathbb{A}^*)$. $\mathcal{B}$ submits (k_i, S_{k_i}) to the challenger. Note that (k_i, S_{k_i}) satisfies the the restriction on $\mathcal{B}$ in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$ that (k_i, S_{k_i}) does not satisfy $(k_i = \bar{k}) \wedge (k_i \in [N] \setminus R^*) \wedge (S_{k_i}\ satisfies\ \mathbb{A}^*)$. $\mathcal{B}$ receives secret key $\mathsf{SK}_{k_i,S_{k_i}}$ from the challenger. Then $\mathcal{B}$ gives $\mathsf{SK}_{k_i,S_{k_i}}$ to $\mathcal{A}$.

Guess. $\mathcal{A}$ outputs a guess $\gamma' \in \{0,1\}$ for γ. If $\gamma' = \gamma$ then $\mathcal{B}$ returns 0 to its challenger. Otherwise $\mathcal{B}$ returns 1 to its challenger.

Now, observe that when $b = 0$ then $\mathcal{B}$ is emulating perfectly an $\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}]$ challenger. When $b = 1$ then $\mathcal{B}$ is emulating perfectly an $\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}+1]$ challenger. A standard argument now shows that $|\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}] - \mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}+1]| \leq 2 \cdot \mathsf{IH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{B}}[\bar{k}]$ as required.

B Correctness

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

$$\frac{e(\bar{K}_{i,j}, Q_i) \cdot e(K''_{i,j}, Q''_i)}{e(K'_{i,j}, Q'_i)} = \frac{e\big(g^{\alpha_i} g^{r_i c_j} (f \prod_{j' \in \bar{R}_i} f_{j'})^{\sigma_{i,j}}, g^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}\big) e(Z_i^{\sigma_{i,j}}, g^{t_i})}{e\big(g^{\sigma_{i,j}}, (f \prod_{j' \in \bar{R}_i} f_{j'})^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)} Z_i^{t_i} f^{\pi}\big)}$$
$$= \frac{e(g^{\alpha_i}, g^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}) e(g^{r_i c_j}, g^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)})}{e\big(g^{\sigma_{i,j}}, f^{\pi}\big)}.$$

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

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

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

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

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

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

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

C Proof of Lemma 1

Proof. Suppose there exists a PPT adversary $\mathcal{A}$ that selectively breaks the index-hiding game with non-negligible advantage $Adv_{\mathcal{A}}$. We construct a PPT algorithm $\mathcal{B}$, which is given a D3DH problem instance and a Modified Source Group q-parallel BDHE problem instance, and solves at least one of the two problems with non-negligible advantage. $\mathcal{B}$ flips a random coin $c \in \{0,1\}$, if $c = 0$, $\mathcal{B}$ interacts with $\mathcal{A}$ in **Case A** as guessing "$\mathcal{A}$ is not in **Case II.3**", otherwise $\mathcal{B}$ interacts with $\mathcal{A}$ in **Case B** as guessing "$\mathcal{A}$ is not in **Case II.1**".

Case A: $\mathcal{B}$ **uses** $\mathcal{A}$ **to solve the D3DH problem.** Garg et al. [9, Sec. 5.1] proposed an AugBE scheme $\Sigma_{\mathsf{AugBE}} = (\mathsf{Setup}_{\mathsf{AugBE}}, \mathsf{Encrypt}_{\mathsf{AugBE}}, \mathsf{Decrypt}_{\mathsf{AugBE}})$ and proved that it is index-hiding. The Lemma 6.2 of [9] states that *if the D3DH assumption holds, then for $\bar{j} < m$ no PPT adversary can distinguish between an encryption to $(\bar{i}, \bar{j})$ and $(\bar{i}, \bar{j}+1)$ in the index-hiding game for Σ_{AugBE} with non-negligible probability.* Note that if $\mathcal{A}$ is in **Case I**, **Case II.1** or **Case II.2**, it also follows the restrictions of the index-hiding game for Σ_{AugBE}, here we do not build a direct reduction that uses $\mathcal{A}$ to solve the D3DH problem, instead, we build a reduction to break the index-hiding property of Σ_{AugBE}. We first give the reduction sketch below.

First we review the structures of public key $\mathsf{PK}^{\mathsf{AugBE}}$, private key $\mathsf{SK}^{\mathsf{AugBE}}_{(i,j)}$ and ciphertext $CT^{\mathsf{AugBE}}_{\bar{R}}$ of Σ_{AugBE} [9][4].

$$\begin{aligned}
\mathsf{PK}^{\mathsf{AugBE}} &= \big(\ g,\ \{E_i = e(g,g)^{\alpha_i},\ G_i = g^{r_i}\}_{i\in[m]},\quad \{H_j = g^{c_j},\ f_j\}_{j\in[m]}\ \big),\\
\mathsf{SK}^{\mathsf{AugBE}}_{(i,j)} &= \big(K_{i,j},\ K'_{i,j},\quad \{\bar{K}_{i,j,j'}\}_{j'\in[m]\setminus\{j\}}\ \big) = \big(g^{\alpha_i} g^{r_i c_j} f_j^{\sigma_{i,j}},\ g^{\sigma_{i,j}},\quad \{f_{j'}^{\sigma_{i,j}}\}_{j'\in[m]\setminus\{j\}}\ \big),\\
CT^{\mathsf{AugBE}}_{\bar{R}} &= \langle (\boldsymbol{R}_i, \boldsymbol{R}'_i, Q_i, Q'_i, T_i)_{i=1}^m,\ (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^m,\ \bar{R}\rangle,
\end{aligned}$$

where $CT^{\mathsf{AugBE}}_{\bar{R}}$ is for receiver list $\bar{R}$ and index (i^*, j^*) with

1. For each $i \in [m]$:
   - if $i < i^*$: $\boldsymbol{R}_i = g^{\boldsymbol{v}_i},\ \ \boldsymbol{R}'_i = g^{\kappa \boldsymbol{v}_i},\ \ Q_i = g^{s_i},\ \ Q'_i = (\prod_{j'\in\bar{R}_i} f_{j'})^{s_i},\ \ T_i = E_i^{\hat{s}_i}$.
   - if $i \geq i^*$: $\boldsymbol{R}_i = G_i^{s_i \boldsymbol{v}_i},\ \boldsymbol{R}'_i = G_i^{\kappa s_i \boldsymbol{v}_i},\ Q_i = g^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)},\ \ Q'_i = (\prod_{j'\in\bar{R}_i} f_{j'})^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)},\ T_i = M \cdot E_i^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}$.
2. For each $j \in [m]$:
   - if $j < j^*$: $\boldsymbol{C}_j = H_j^{\tau(\boldsymbol{v}_c + \mu_j \boldsymbol{\chi}_3)} \cdot g^{\kappa \boldsymbol{w}_j},\ \ \boldsymbol{C}'_j = g^{\boldsymbol{w}_j}$.
   - if $j \geq j^*$: $\boldsymbol{C}_j = H_j^{\tau \boldsymbol{v}_c} \cdot g^{\kappa \boldsymbol{w}_j},\ \ \boldsymbol{C}'_j = g^{\boldsymbol{w}_j}$.

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

Setup. From the received $\mathsf{PK}^{\mathsf{AugBE}}$, $\mathcal{B}$ generates PP for $\mathcal{A}$ by randomly choosing β, θ, z_i $(i \in [m]) \in \mathbb{Z}_p$ and $h, H \in \mathbb{G}$, and setting $f = g^{\beta}, G = g^{\theta}, \{Z_i = g^{z_i}\}_{i\in[m]}$.
Phase 1 and **2**. As $\mathcal{B}$ can compute $f^{\sigma_{i,j}} = (g^{\sigma_{i,j}})^{\beta}$, $Z_i^{\sigma_{i,j}} = (g^{\sigma_{i,j}})^{z_i}$, and $G^{-\sigma_{i,j}} = (g^{\sigma_{i,j}})^{-\theta}$ without $\sigma_{i,j}$, $\mathcal{B}$ can produce $\mathsf{SK}_{(i,j),S_{(i,j)}}$ for $\mathcal{A}$, using $\mathsf{SK}^{\mathsf{AugBE}}_{(i,j)}$ and random $\{\delta_{i,j,x}\}_{x\in S_{(i,j)}}$.
Challenge. As $\mathcal{B}$ can compute $f^{s_i} = (g^{s_i})^{\beta}$ and $f^{\tau s_i(\boldsymbol{v_i}\cdot\boldsymbol{v}_c)} = (g^{\tau s_i(\boldsymbol{v_i}\cdot\boldsymbol{v}_c)})^{\beta}$ without s_i or $\tau s_i(\boldsymbol{v_i}\cdot\boldsymbol{v}_c)$, by using its challenge ciphertext $CT^{\mathsf{AugBE}}_{\bar{R}^*}$ (for $\bar{R}^* = [m,m]\setminus R^*$) and random $t_i (i \in [m]), \xi_k (k \in [l]) \in \mathbb{Z}_p, \boldsymbol{u} = (\pi, u_2, \ldots, u_n) \in \mathbb{Z}_p^n$, $\mathcal{B}$ can produce the challenge ciphertext $CT_{R^*,(A^*,\rho^*)}$ for $\mathcal{A}$.
Guess. $\mathcal{B}$ sends $\mathcal{A}$'s guess $b' \in \{0,1\}$ to its challenger.

During the interaction, if $\mathcal{A}$ is in **Case II.3**, $\mathcal{B}$ will abort and return a random $b \in \{0,1\}$ to its challenger.
Now we give the reduction details.

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

Setup. The challenger gives $\mathcal{B}$ the public key $\mathsf{PK}^{\mathsf{AugBE}}$

$$\mathsf{PK}^{\mathsf{AugBE}} = \big(\ g,\ \{E_i = e(g,g)^{\alpha_i},\ G_i = g^{r_i}\}_{i\in[m]},\ \ \{H_j = g^{c_j},\ f_j\}_{j\in[m]}\ \big),$$

and private keys $\{\mathsf{SK}^{\mathsf{AugBE}}_{(i,j)}\}_{(i,j)\in[m,m]\setminus\{(\bar{i},\bar{j})\}}$ as

$$\mathsf{SK}^{\mathsf{AugBE}}_{(i,j)} = \big(\tilde{K}_{i,j},\ \tilde{K}'_{i,j},\ \ \{\tilde{\tilde{K}}_{i,j,j'}\}_{j'\in[m]\setminus\{j\}}\ \big) = \big(g^{\alpha_i} g^{r_i c_j} f_j^{\sigma_{i,j}},\ g^{\sigma_{i,j}},\ \ \{f_{j'}^{\sigma_{i,j}}\}_{j'\in[m]\setminus\{j\}}\ \big),$$

where $g, f_1, \ldots, f_m \in \mathbb{G}$ and $\{\alpha_i, r_i \in \mathbb{Z}_p\}_{i\in[m]}$, $\{c_j \in \mathbb{Z}_p\}_{j\in[m]}$, $\{\sigma_{i,j} \in \mathbb{Z}_p\}_{(i,j)\in[m,m]\setminus\{(\bar{i},\bar{j})\}}$ are randomly chosen. $\mathcal{B}$ sets $\tilde{c} = 0$ to denote that $\mathcal{B}$ does not obtain the private $\mathsf{SK}^{\mathsf{AugBE}}_{(\bar{i},\bar{j})}$.

$\mathcal{B}$ randomly chooses $\beta, \theta,\ z_1, \ldots, z_m \in \mathbb{Z}_p$ and $h, H \in \mathbb{G}$, then gives $\mathcal{A}$ the following public parameter PP:

$$\mathsf{PP} = \big(g,\ h,\ f = g^{\beta},\ f_1, \ldots, f_m,\ G = g^{\theta},\ H,\ \ \{E_i,\ G_i,\ Z_i = g^{z_i}\}_{i\in[m]},\ \ \{H_j\}_{j\in[m]}\big).$$

Phase 1. $\mathcal{A}$ adaptively submits $((i,j), S_{(i,j)})$ to $\mathcal{B}$. If $(i,j) = (\bar{i},\bar{j})$, then $\mathcal{B}$ sets $\tilde{c} = 1$ and submits $\tilde{c}$ to its challenger, and receives the private key $\mathsf{SK}^{\mathsf{AugBE}}_{(\bar{i},\bar{j})}$. $\mathcal{B}$ randomly chooses $\delta_{i,j,x} \in \mathbb{Z}_p\ \forall x \in S_{(i,j)}$, then creates the private key $\mathsf{SK}_{(i,j),S_{(i,j)}} = \big(\ (i,j), S_{(i,j)},\ K_{i,j}, K'_{i,j}, K''_{i,j},\ \ \{\bar{K}_{i,j,j'}\}_{j'\in[m]\setminus\{j\}},$ $\{K_{i,j,x}, K'_{i,j,x}\}_{x\in S_{(i,j)}}\ \big)$ from $\mathsf{SK}^{\mathsf{AugBE}}_{(i,j)}$ as

$$\begin{aligned}
&K_{i,j} = \tilde{K}_{i,j}\cdot(\tilde{K}'_{i,j})^{\beta},\ \ K'_{i,j} = \tilde{K}'_{i,j},\ \ K''_{i,j} = (\tilde{K}'_{i,j})^{z_i},\ \ \ \{\bar{K}_{i,j,j'} = \tilde{\tilde{K}}_{i,j,j'}\}_{j'\in[m]\setminus\{j\}},\\
&\{K_{i,j,x} = g^{\delta_{i,j,x}},\ K_{i,j,x} = (H^x h)^{\delta_{i,j,x}}(\tilde{K}'_{i,j})^{-\theta}\}_{x\in S_{(i,j)}}.
\end{aligned}$$

Challenge. $\mathcal{A}$ submits a message M and a revocation list R^*. $\mathcal{B}$ sets $\bar{R}^* = [m,m]\setminus R^*$.

- if $(\bar{i},\bar{j}) \in \bar{R}^* \wedge \tilde{c} = 1$: $\mathcal{A}$ is in **Case II.3**. $\mathcal{B}$ returns a random $\beta_3 \in \{0,1\}$ to its challenger, then aborts.
- if $(\bar{i},\bar{j}) \in \bar{R}^* \wedge \tilde{c} = 0$: $\mathcal{B}$ continues the following interaction.
- if $(\bar{i},\bar{j}) \notin \bar{R}^* \wedge \tilde{c} = 1$: $\mathcal{B}$ continues the following interaction.
- if $(\bar{i},\bar{j}) \notin \bar{R}^* \wedge \tilde{c} = 0$: $\mathcal{B}$ sets $\tilde{c} = 1$ and submits c_1 to its challenger, and receives the private key $\mathsf{SK}^{\mathsf{AugBE}}_{(\bar{i},\bar{j})}$. Then $\mathcal{B}$ continues the following interaction.

Now $\mathcal{B}$ ends the Query Phase for the AugBE index-hiding game with its challenger, and submits $(M, \bar{R}^*)$ to the challenger. Note that from the view of the challenger, $\mathcal{B}$'s behaviors satisfy the restrictions in the AugBE index-hiding game, i.e., if $\mathcal{B}$ sends $\tilde{c} = 1$ to the challenger and obtains $\mathsf{SK}_{(\bar{i},\bar{j})}^{\mathsf{AugBE}}$ then $(\bar{i}, \bar{j}) \notin \bar{R}^*$. The challenger gives $\mathcal{B}$ the challenge ciphertext $CT_{\bar{R}^*}^{\mathsf{AugBE}} = \langle (\tilde{\boldsymbol{R}}_i, \tilde{\boldsymbol{R}}'_i, \tilde{Q}_i, \tilde{Q}'_i, \tilde{T}_i)_{i=1}^m, (\tilde{\boldsymbol{C}}_j, \tilde{\boldsymbol{C}}'_j)_{j=1}^m, \bar{R}^* \rangle$, which is encrypted to $(i^*, j^*) \in \{(\bar{i}, \bar{j}), (\bar{i}, \bar{j}+1)\}$ and in the form of

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

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

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

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

Phase 2. $\mathcal{A}$ adaptively submits $((i, j), S_{(i,j)})$ to $\mathcal{B}$.

- if $(i, j) \neq (\bar{i}, \bar{j})$: $\mathcal{B}$ creates the private key $\mathsf{SK}_{(i,j),S_{(i,j)}}$ from $\mathsf{SK}_{(i,j)}^{\mathsf{AugBE}}$ as in **Phase 1**.
- if $(i, j) = (\bar{i}, \bar{j}) \wedge \tilde{c} = 1$: this implies $\mathcal{B}$ has obtained $\mathsf{SK}_{(\bar{i},\bar{j})}^{\mathsf{AugBE}}$ from its challenger. $\mathcal{B}$ creates the private key $\mathsf{SK}_{(\bar{i},\bar{j}),S_{(\bar{i},\bar{j})}}$ from $\mathsf{SK}_{(\bar{i},\bar{j})}^{\mathsf{AugBE}}$ as in **Phase 1**.
- if $(i, j) = (\bar{i}, \bar{j}) \wedge \tilde{c} = 0$: observing $\mathcal{B}$'s behaviors in **Challenge** phase, we have that $\tilde{c} = 0$ implies $(\bar{i}, \bar{j}) \in \bar{R}^*$. In other words, $\mathcal{A}$ is querying a key with index $(\bar{i}, \bar{j})$ and $(\bar{i}, \bar{j}) \in \bar{R}^*$, i.e., $\mathcal{A}$ is in **Case II.3**. $\mathcal{B}$ return a random $\beta_3 \in \{0, 1\}$ to its challenger, then aborts.

Guess. $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ to $\mathcal{B}$, then $\mathcal{B}$ sets $\beta_3 = b'$ ands returns β_3 to its challenger.

When $\mathcal{B}$ does not abort, $\mathcal{B}$'s advantage in the index-hiding game for Σ_{AugBE} will be exactly equal to $\mathcal{A}$'s advantage in the index-hiding game for our AugR-CP-ABE scheme Σ_{A}. Thus, $\mathcal{B}$'s final advantage in the index-hiding game for Σ_{AugBE} is $Adv_{\mathcal{B},3} = Adv_{\mathcal{A}} \cdot \Pr[\mathcal{A} \text{ is not in } \textbf{Case II.3} \wedge (c = 0)]$.

Case B: **$\mathcal{B}$ uses $\mathcal{A}$ to solve the Modified Source Group q-parallel BDHE problem.** $\mathcal{B}$ is given

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

and T, where $(p, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}$, $g \xleftarrow{R} \mathbb{G}$, $a, c, d, b_1, \ldots, b_q \xleftarrow{R} \mathbb{Z}_p$, and T is either equal to $g^{ca^{q+1}}$ or is a random element of $\mathbb{G}$. $\mathcal{B}$'s goal is to determine $T = g^{ca^{q+1}}$ or T is a random element from $\mathbb{G}$.

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

Setup. $\mathcal{B}$ randomly chooses $\{\alpha_i \in \mathbb{Z}_p\}_{i\in[m]}$, $\{r_i,\ z'_i \in \mathbb{Z}_p\}_{i\in[m]\setminus\{\bar{i}\}}$, $r'_{\bar{i}}$, $z_{\bar{i}}$, $\{c'_j \in \mathbb{Z}_p\}_{j\in[m]}$, and $\beta, \theta, \eta, \theta_1, \ldots, \theta_m \in \mathbb{Z}_p$. $\mathcal{B}$ gives $\mathcal{A}$ the public parameter PP:

$$\Big(g,\ f = g^a,\ \ \{f_j = g^{\theta_j}\}_{j\in[m]},\ \ h = g^\beta \cdot \prod_{k\in[l]} \prod_{t\in[n]} \big(g^{a^t/b_k^2}\big)^{-\rho^*(k)A^*_{k,t}},$$
$$G = g^\theta \cdot \prod_{k\in[l]} \prod_{t\in[n]} \big(g^{a^t/b_k}\big)^{A^*_{k,t}},\ \ H = g^\eta \cdot \prod_{k\in[l]} \prod_{t\in[n]} \big(g^{a^t/b_k^2}\big)^{A^*_{k,t}},\ \ \{E_i = e(g,g)^{\alpha_i}\}_{i\in[m]},$$
$$\{G_i = g^{r_i},\ Z_i = (g^a)^{z'_i}\}_{i\in[m]\setminus\{\bar{i}\}},\ G_{\bar{i}} = (g^{a^q})^{r'_{\bar{i}}},\ Z_{\bar{i}} = g^{z_{\bar{i}}},$$
$$\{H_j = (g^d)^{c'_j}\}_{j\in[m]\setminus\{\bar{j}\}},\ H_{\bar{j}} = (g^a)^{c'_{\bar{j}}}\ \Big).$$

Note that $\mathcal{B}$ implicitly chooses $r_{\bar{i}} \in \mathbb{Z}_p$, $\{z_i \in \mathbb{Z}_p\}_{i\in[m]\setminus\{\bar{i}\}}$, $\{c_j \in \mathbb{Z}_p\}_{j\in[m]}$ such that

$$a^q r'_{\bar{i}} \equiv r_{\bar{i}} \bmod p,\ \ a z'_i \equiv z_i \bmod p\ \forall i \in [m] \setminus \{\bar{i}\},$$
$$d c'_j \equiv c_j \bmod p\ \forall j \in [m] \setminus \{\bar{j}\},\ \ a c'_{\bar{j}} \equiv c_{\bar{j}} \bmod p.$$

Phase 1. To respond to $\mathcal{A}$'s query for $((i,j), S_{(i,j)})$,

• if $(i,j) \neq (\bar{i}, \bar{j})$: $\mathcal{B}$ chooses $\sigma_{i,j} \in \mathbb{Z}_p$, $\{\delta_{i,j,x} \in \mathbb{Z}_p\}_{x\in S_{(i,j)}}$ at random, then creates the decryption key $\mathsf{SK}_{(i,j),S_{(i,j)}}$:

$$K_{i,j} = \begin{cases} g^{\alpha_i}(g^d)^{r_i c'_j}(ff_j)^{\sigma_{i,j}}, & : i \neq \bar{i}, j \neq \bar{j} \\ g^{\alpha_i}(g^{da^q})^{r'_i c'_j}(ff_j)^{\sigma_{i,j}}, & : i = \bar{i}, j \neq \bar{j} \\ g^{\alpha_i}(g^a)^{r_i c'_{\bar{j}}}(ff_j)^{\sigma_{i,j}}, & : i \neq \bar{i}, j = \bar{j} \end{cases}$$
$$K'_{i,j} = g^{\sigma_{i,j}},\ K''_{i,j} = Z_i^{\sigma_{i,j}},\ \ \{\bar{K}_{i,j,j'} = f_{j'}^{\sigma_{i,j}}\}_{j'\in[m]\setminus\{j\}},$$
$$\{K_{i,j,x} = g^{\delta_{i,j,x}},\ K'_{i,j,x} = (H^x h)^{\delta_{i,j,x}} G^{-\sigma_{i,j}}\}_{x\in S_{(i,j)}}.$$

• if $(i,j) = (\bar{i}, \bar{j})$: if $S_{(i,j)}$ satisfies (A^*, ρ^*), then $\mathcal{A}$ is in **Case II.1**, $\mathcal{B}$ returns a random $\beta_q \in \{0,1\}$ to the challenger. Otherwise (i.e. $S_{(i,j)}$ does not satisfy (A^*, ρ^*)), $\mathcal{B}$ first computes a vector $\bar{\boldsymbol{u}} = (\bar{u}_1, \ldots, \bar{u}_n) \in \mathbb{Z}_p^n$ that has first entry equal to $-r'_{\bar{i}} c'_{\bar{j}}$ (i.e. $\bar{u}_1 = -r'_{\bar{i}} c'_{\bar{j}}$) and is orthogonal to all of the rows A^*_k of A^* such that $\rho^*(k) \in S_{(i,j)}$ (i.e. $A^*_k \cdot \bar{\boldsymbol{u}} = 0\ \forall k \in [l]\ s.t.\ \rho^*(k) \in S_{(i,j)}$). Note that such a vector must exist since $S_{(i,j)}$ fails to satisfy (A^*, ρ^*), and it is efficiently computable. Then $\mathcal{B}$ randomly chooses $\sigma'_{\bar{i},\bar{j}} \in \mathbb{Z}_p$, $\{\delta'_{\bar{i},\bar{j},x} \in \mathbb{Z}_p\}_{x\in S_{(i,j)}}$ and sets the values of $\sigma_{\bar{i},\bar{j}}$ and $\{\delta_{\bar{i},\bar{j},x}\}_{x\in S_{(i,j)}}$ by implicitly setting

$$\sigma_{\bar{i},\bar{j}} = \sigma'_{\bar{i},\bar{j}} + \sum_{t\in[n]} \bar{u}_t a^{q+1-t}, \tag{2}$$

$$\delta_{\bar{i},\bar{j},x} = \delta'_{\bar{i},\bar{j},x} + \sigma'_{\bar{i},\bar{j}} \cdot \sum_{\substack{k'\in[l] \\ \rho^*(k')\notin S_{(i,j)}}} \frac{b_{k'}}{x - \rho^*(k')} + \sum_{\substack{k'\in[l] \\ \rho^*(k')\notin S_{(i,j)}}} \sum_{t\in[n]} \frac{\bar{u}_t b_{k'} a^{q+1-t}}{x - \rho^*(k')}. \tag{3}$$

Note that for $x \in S_{(i,j)}$ and $\rho^*(k') \notin S_{(i,j)}$ we have $x - \rho^*(k') \neq 0$. $\mathcal{B}$ creates the private key $\mathsf{SK}_{(\bar{i},\bar{j}),S_{(\bar{i},\bar{j})}}$ as follows:

$$K_{\bar{i},\bar{j}} = g^{\alpha_{\bar{i}}} f^{\sigma'_{\bar{i},\bar{j}}} \big(\prod_{t=2}^{n} (g^{a^{q-t+2}})^{\bar{u}_t}\big) \big(g^{\sigma'_{\bar{i},\bar{j}}} \prod_{t=1}^{n} (g^{a^{q-t+1}})^{\bar{u}_t}\big)^{\theta_{\bar{j}}}, \quad K'_{\bar{i},\bar{j}} = g^{\sigma'_{\bar{i},\bar{j}}} \prod_{t=1}^{n} (g^{a^{q-t+1}})^{\bar{u}_t}, \quad K''_{\bar{i},\bar{j}} = (K'_{\bar{i},\bar{j}})^{z_{\bar{i}}},$$
$$\{\bar{K}_{\bar{i},\bar{j},j'} = \big(g^{\sigma'_{\bar{i},\bar{j}}} \prod_{t=1}^{n} (g^{a^{q-t+1}})^{\bar{u}_t}\big)^{\theta_{j'}}\}_{j' \in [m] \setminus \{\bar{j}\}}.$$

For $x \in S_{(i,j)}$, we have

$$K_{\bar{i},\bar{j},x} = g^{\delta_{\bar{i},\bar{j},x}} = g^{\delta'_{\bar{i},\bar{j},x}} \big(\prod_{\substack{k' \in [l] \\ \rho^*(k') \notin S_{(i,j)}}} (g^{b_{k'}})^{\sigma'_{\bar{i},\bar{j}}/(x-\rho^*(k'))} \big) \cdot \big(\prod_{\substack{k' \in [l] \\ \rho^*(k') \notin S_{(i,j)}}} \prod_{t \in [n]} (g^{b_{k'} a^{q+1-t}})^{\bar{u}_t/(x-\rho^*(k'))} \big),$$

and after some algebraic manipulations (the details are given in Appendix C.1), we have

$$(H^x h)^{\delta_{\bar{i},\bar{j},x}} = \Psi_1 \cdot \big(\prod_{\substack{k \in [l] \\ \rho^*(k) \notin S_{(i,j)}}} \prod_{t \in [n]} \prod_{t' \in [n]} (g^{a^{q+1-t'+t}/b_k})^{A^*_{k,t} \bar{u}_{t'}} \big),$$
$$G^{-\sigma_{\bar{i},\bar{j}}} = \Psi_2 \cdot \big(\prod_{k \in [l]} \prod_{t \in [n]} \prod_{t' \in [n]} (g^{a^{q+1-t'+t}/b_k})^{-A^*_{k,t} \bar{u}_{t'}} \big),$$

where Ψ_1 and Ψ_2 can be calculated using the suitable terms of the assumption. Thus, we have

$$\begin{aligned}
K'_{\bar{i},\bar{j},x} &= (H^x h)^{\delta_{\bar{i},\bar{j},x}} G^{-\sigma_{\bar{i},\bar{j}}} \\
&= \Psi_1 \cdot \Psi_2 \cdot \big(\prod_{\substack{k \in [l] \\ \rho^*(k) \in S_{(i,j)}}} \prod_{t \in [n]} \prod_{t' \in [n]} (g^{a^{q+1-t'+t}/b_k})^{-A^*_{k,t} \bar{u}_{t'}} \big) \\
&= \Psi_1 \cdot \Psi_2 \cdot \underbrace{\big(\prod_{\substack{k \in [l] \\ \rho^*(k) \in S_{(i,j)}}} \prod_{t \in [n]} \prod_{t' \in [n] \setminus \{t\}} (g^{a^{q+1-t'+t}/b_k})^{-A^*_{k,t} \bar{u}_{t'}} \big)}_{\Psi_3 \quad (for\ t' \neq t)} \cdot \underbrace{\big(\prod_{\substack{k \in [l] \\ \rho^*(k) \in S_{(i,j)}}} \prod_{t \in [n]} (g^{a^{q+1}/b_k})^{-A^*_{k,t} \bar{u}_t} \big)}_{for\ t'=t} \\
&= \Psi_1 \cdot \Psi_2 \cdot \Psi_3 \cdot \big(\prod_{\substack{k \in [l] \\ \rho^*(k) \in S_{(i,j)}}} (g^{a^{q+1}/b_k})^{-(A^*_k \cdot \bar{\boldsymbol{u}})} \big) \\
&= \Psi_1 \cdot \Psi_2 \cdot \Psi_3, \quad (\text{since } A^*_k \cdot \bar{\boldsymbol{u}} = 0 \ \forall k \in [l] \ s.t. \ \rho^*(k) \in S_{(i,j)})
\end{aligned}$$

Note that Ψ_1, Ψ_2 and Ψ_3 can be calculated using the suitable terms of the assumption, $\mathcal{B}$ can calculate $K'_{\bar{i},\bar{j},x}$.

Challenge. $\mathcal{A}$ submits a message M and a revocation list R^*. $\mathcal{B}$ randomly chooses

$$\begin{aligned}
\tau', \quad s_1, \ldots, s_{\bar{i}-1}, s'_{\bar{i}}, s_{\bar{i}+1}, \ldots, s_m &\in \mathbb{Z}_p, \\
t'_1, \ldots, t'_{\bar{i}-1}, t_{\bar{i}}, t'_{\bar{i}+1}, \ldots, t'_m &\in \mathbb{Z}_p, \\
\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{\bar{j}-1}, \boldsymbol{w}'_{\bar{j}}, \ldots, \boldsymbol{w}'_m &\in \mathbb{Z}_p^3,
\end{aligned}$$

$$\xi'_1, \ldots, \xi'_l,\ \pi' \in \mathbb{Z}_p,\quad \boldsymbol{u}' = (0, u'_2, \ldots, u'_n)\ \in \mathbb{Z}_p^n.$$

$\mathcal{B}$ randomly chooses $r_x, r_y, r_z \in \mathbb{Z}_p$, and sets $\boldsymbol{\chi}_1 = (r_x, 0, r_z)$, $\boldsymbol{\chi}_2 = (0, r_y, r_z), \boldsymbol{\chi}_3 = \boldsymbol{\chi}_1 \times \boldsymbol{\chi}_2$, then randomly chooses

$$\begin{aligned}
&\boldsymbol{v}_i \in \mathbb{Z}_p^3\ \forall i \in \{1, \ldots, \bar{i} - 1\},\\
&\boldsymbol{v}_{\bar{i}}^p \in span\{\boldsymbol{\chi}_1, \boldsymbol{\chi_2}\},\ \boldsymbol{v}_{\bar{i}}^q \in span\{\boldsymbol{\chi}_3\},\\
&\boldsymbol{v}_i \in span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}\ \forall i \in \{\bar{i} + 1, \ldots, m\},\\
&\boldsymbol{v}_c^p \in span\{\boldsymbol{\chi}_1, \boldsymbol{\chi_2}\},\ \boldsymbol{v}_c^q = \nu_c \boldsymbol{\chi}_3 \in span\{\boldsymbol{\chi}_3\}.
\end{aligned}$$

$\mathcal{B}$ sets the values of $\kappa, \tau,\ s_{\bar{i}},\ t_i (i \in [m] \setminus \{\bar{i}\}) \in \mathbb{Z}_p$, $\boldsymbol{v}_{\bar{i}}$, $\boldsymbol{v}_c$, $\boldsymbol{w}_j (j \in \{\bar{j}, \ldots, m\}) \in \mathbb{Z}_p^3$, $\pi \in \mathbb{Z}_p$, $\boldsymbol{u} \in \mathbb{Z}_p^n$, and $\{\xi_k \in \mathbb{Z}_p\}_{k \in [l]}$ by implicitly setting

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

It is worth noticing that $\boldsymbol{v}_{\bar{i}}$ and $\boldsymbol{v}_c$ are random vectors in $\mathbb{Z}_p^3$ as required, and $(\boldsymbol{v}_{\bar{i}} \cdot \boldsymbol{v}_c) = \frac{1}{c}(\boldsymbol{v}_{\bar{i}}^p \cdot \boldsymbol{v}_c^p) + d(\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)$, since $\boldsymbol{\chi}_3$ is orthogonal to $span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$ and $\mathbb{Z}_p^3 = span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2, \boldsymbol{\chi}_3\}$.

Let $\bar{R}^* = [m, m] \backslash R^*$ and $\bar{R}_i^* = \{j' | (i, j') \in \bar{R}^*\}\ \forall i \in [m]$. $\mathcal{B}$ creates the ciphertext $\langle R^*, (A^*, \rho^*),\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, Q_i, Q'_i, Q''_i, T_i)_{i=1}^m,\ (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^m,\ (P_k, P'_k, P''_k)_{k=1}^l \rangle$ as follows:

1. For each row $i \in [m]$:

– if $i < \bar{i}$: it randomly chooses $\hat{s}_i \in \mathbb{Z}_p$, then sets

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

– if $i = \bar{i}$: it sets

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

– if $i > \bar{i}$: it sets

$$\boldsymbol{R}_i = g^{r_i s_i \boldsymbol{v}_i},\quad \boldsymbol{R}'_i = (g^{a^q})^{r_i s_i \boldsymbol{v}_i},$$

$$Q_i = (g^{a^q})^{\tau' s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c^p)},\ Q_i' = (\prod_{j' \in \bar{R}_i^*} Q_i^{\theta_{j'}}) Z_i^{t_i'} f^{\pi'},\ Q_i'' = g^{t_i'} (g^{a^q})^{-\tau' s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c^p)/z_i'} (g^{cd})^{\tau' s_i' (\boldsymbol{v}_i^q \cdot \boldsymbol{v}_c^q)/z_i'},$$

$$T_i = M \cdot e(g^{\alpha_i}, Q_i).$$

2. For each $j \in [m]$:

– if $j < \bar{j}$: it randomly chooses $\mu_j' \in \mathbb{Z}_p$ and implicitly sets the value of μ_j such that $(\mu_j'/(cda^q) - 1)\nu_c \equiv \mu_j \bmod p$, then sets: $\boldsymbol{C}_j = (g^{da^q})^{c_j' \tau' \boldsymbol{v}_c^p} \cdot g^{c_j' \tau' \mu_j' \boldsymbol{v}_c^q} \cdot (g^{a^q})^{\boldsymbol{w}_j},\ \ \boldsymbol{C}_j' = g^{\boldsymbol{w}_j}$.

– if $j = \bar{j}$: $\boldsymbol{C}_j = T^{c_{\bar{j}}' \tau' \boldsymbol{v}_c^q} \cdot (g^{a^q})^{\boldsymbol{w}_j'},\ \ \boldsymbol{C}_j' = g^{\boldsymbol{w}_{\bar{j}}'} \cdot (g^a)^{-c_{\bar{j}}' \tau' \boldsymbol{v}_c^p}$.

– if $j > \bar{j}$: $\boldsymbol{C}_j = (g^{da^q})^{c_j' \tau' \boldsymbol{v}_c^p} \cdot (g^{a^q})^{\boldsymbol{w}_j'}, \boldsymbol{C}_j' = g^{\boldsymbol{w}_j'} \cdot (g^{cd})^{-c_j' \tau' \boldsymbol{v}_c^q}$.

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

$$\begin{aligned}
P_k &= f^{A_k^* \cdot \boldsymbol{u}} G^{\xi_k} = \big(f^{A_k^* \cdot (1, a, \ldots, a^{n-1})}\big)^{\pi} \cdot \underbrace{f^{A_k^* \cdot \boldsymbol{u}'} G^{\xi_k'}}_{\Phi_1} \cdot \big(g^{\theta} \prod_{k' \in [l]} \prod_{t \in [n]} (g^{a^t/b_{k'}})^{A_{k',t}^*}\big)^{cdb_k \tau' s_i' (\boldsymbol{v}_i^q \cdot \boldsymbol{v}_c^q)} \\
&= \big(\prod_{t \in [n]} (g^{a^t})^{A_{k,t}^*}\big)^{\pi' - cd\tau' s_{\bar{i}}' (\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)} \cdot \Phi_1 \cdot \underbrace{\big(g^{cdb_k}\big)^{\theta \tau' s_{\bar{i}}' (\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)}}_{\Phi_2} \cdot \big(\prod_{k' \in [l]} \prod_{t \in [n]} (g^{cda^t b_k / b_{k'}})^{A_{k',t}^*}\big)^{\tau' s_{\bar{i}}' (\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)} \\
&= \underbrace{\big(\prod_{t \in [n]} (g^{a^t})^{A_{k,t}^*}\big)^{\pi'}}_{\Phi_3} \cdot \underbrace{\big(\prod_{t \in [n]} (g^{cda^t})^{A_{k,t}^*}\big)^{-\tau' s_{\bar{i}}' (\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)}}_{\Delta} \cdot \Phi_1 \cdot \Phi_2 \\
&\quad \cdot \underbrace{\big(\prod_{k' \in [l] \setminus \{k\}} \prod_{t \in [n]} (g^{cda^t b_k / b_{k'}})^{A_{k',t}^*}\big)^{\tau' s_{\bar{i}}' (\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)}}_{\Phi_4 \quad (for\ k' \neq k)} \cdot \underbrace{\big(\prod_{t \in [n]} (g^{cda^t})^{A_{k,t}^*}\big)^{\tau' s_{\bar{i}}' (\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)}}_{\Delta^{-1} \quad (for\ k' = k)} \\
&= \Phi_3 \cdot \Phi_1 \cdot \Phi_2 \cdot \Phi_4, \\
P_k' &= (H^{\rho^*(k)} h)^{-\xi_k} \\
&= (H^{\rho^*(k)} h)^{-\xi_k'} \cdot \big(g^{\eta \rho^*(k) + \beta}\big)^{-cdb_k \tau' s_{\bar{i}}' (\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)} \cdot \big(\prod_{k' \in [l]} \prod_{t \in [n]} \big(g^{a^t / b_{k'}^2}\big)^{(\rho^*(k) - \rho^*(k')) A_{k',t}^*}\big)^{-cdb_k \tau' s_{\bar{i}}' (\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)} \\
&= \underbrace{(H^{\rho^*(k)} h)^{-\xi_k'} \cdot \big(g^{cdb_k}\big)^{-(\eta \rho^*(k) + \beta) \tau' s_{\bar{i}}' (\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)}}_{\Phi_5} \cdot \big(\prod_{k' \in [l]} \prod_{t \in [n]} \big(g^{cda^t b_k / b_{k'}^2}\big)^{(\rho^*(k') - \rho^*(k)) A_{k',t}^*}\big)^{\tau' s_{\bar{i}}' (\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)} \\
&= \Phi_5 \cdot \underbrace{\big(\prod_{k' \in [l] \setminus \{k\}} \prod_{t \in [n]} \big(g^{cda^t b_k / b_{k'}^2}\big)^{(\rho^*(k') - \rho^*(k)) A_{k',t}^*}\big)^{\tau' s_{\bar{i}}' (\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)}}_{\Phi_6 \quad (for\ k' \neq k)} \\
&\qquad \cdot \underbrace{\big(\prod_{t \in [n]} \big(g^{cda^t b_k / b_k^2}\big)^{(\rho^*(k) - \rho^*(k)) A_{k,t}^*}\big)^{\tau' s_{\bar{i}}' (\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)}}_{1 \quad (for\ k' = k)} \\
&= \Phi_5 \cdot \Phi_6, \\
P_k'' &= g^{\xi_k} = g^{\xi_k'} (g^{cdb_k})^{\tau' s_{\bar{i}}' (\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)}.
\end{aligned}$$

Note that $\Phi_1, \ldots, \Phi_6$ can be calculated using the suitable terms of the assumption, $\mathcal{B}$ can calculate P_k, P_k' and P_k''. If $T = g^{ca^{q+1}}$, the ciphertext is a well-formed encryption to the index $(\bar{i}, \bar{j})$. If T

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

Phase 2. Same as **Phase 1.**

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

When $\mathcal{B}$ does not abort, the distributions of the public parameter, private keys and challenge ciphertext are the same as in the real scheme, $\mathcal{B}$'s advantage in the Modified Source Group q-parallel BDHE game will be exactly equal to $\mathcal{A}$'s advantage in the selective index-hiding game. Thus, $\mathcal{B}$'s final advantage is $Adv_{\mathcal{B},q} = Adv_{\mathcal{A}} \cdot \Pr[\mathcal{A} \text{ is not in } \textbf{Case II.1} \wedge (c=1)]$.

Note that in both **Case A** and **Case B**, the distributions of the public parameter, private keys and challenge ciphertext that $\mathcal{B}$ gives $\mathcal{A}$ are the same as in the real scheme and independent of the value of c. This implies that the value of c and the case that $\mathcal{A}$ is in are independent of each other. Let $\mathcal{A}.\textbf{I}, \mathcal{A}.\textbf{II.1}, \mathcal{A}.\textbf{II.2}$, and $\mathcal{A}.\textbf{II.3}$ be the events that $\mathcal{A}$ is in **Case I**, **Case II.1**, **Case II.2** and **Case II.3**, respectively, and $\overline{\mathcal{A}.\textbf{II.1}}$ and $\overline{\mathcal{A}.\textbf{II.3}}$ be the events that "$\mathcal{A}$ is not in **Case II.1**" and "$\mathcal{A}$ is not in **Case II.3**", respectively. We have

$$\begin{aligned}
Adv_{\mathcal{B},3} + Adv_{\mathcal{B},q} &= Adv_{\mathcal{A}} \cdot \Pr[\overline{\mathcal{A}.\textbf{II.3}} \wedge (c=0)] + Adv_{\mathcal{A}} \cdot \Pr[\overline{\mathcal{A}.\textbf{II.1}} \wedge (c=1)] \\
&= Adv_{\mathcal{A}} \cdot \Pr[\overline{\mathcal{A}.\textbf{II.3}}] \cdot \Pr[c=0] + Adv_{\mathcal{A}} \cdot \Pr[\overline{\mathcal{A}.\textbf{II.1}}] \cdot \Pr[c=1] \\
&= Adv_{\mathcal{A}} \cdot (1 - \Pr[\mathcal{A}.\textbf{II.3}]) \cdot \frac{1}{2} + Adv_{\mathcal{A}} \cdot (1 - \Pr[\mathcal{A}.\textbf{II.1}]) \cdot \frac{1}{2} \\
&= \frac{1}{2} \cdot Adv_{\mathcal{A}} \cdot (2 - (\Pr[\mathcal{A}.\textbf{II.3}] + \Pr[\mathcal{A}.\textbf{II.1}])) \\
&\geq \frac{1}{2} \cdot Adv_{\mathcal{A}},
\end{aligned}$$

since $\Pr[\mathcal{A}.\textbf{II.3}] + \Pr[\mathcal{A}.\textbf{II.1}] \leq \Pr[\mathcal{A}.\textbf{I}] + \Pr[\mathcal{A}.\textbf{II.1}] + \Pr[\mathcal{A}.\textbf{II.2}] + \Pr[\mathcal{A}.\textbf{II.3}] = 1$. This implies that either $Adv_{\mathcal{B},3} \geq \frac{1}{4} \cdot Adv_{\mathcal{A}}$ or $Adv_{\mathcal{B},q} \geq \frac{1}{4} \cdot Adv_{\mathcal{A}}$.

C.1 The Algebraic Manipulation for $K'_{\bar{i},\bar{j},x}$ in Case B of Proof of Lemma 1

For $(i,j) = (\bar{i},\bar{j})$, with the values of $\sigma_{\bar{i},\bar{j}}$ in Equation (2) and the values of $\delta_{\bar{i},\bar{j},x} (\forall x \in S_{(i,j)})$ in Equation (3), for $x \in S_{(i,j)}$, we have

$$\begin{aligned}
(H^x h)^{\delta_{\bar{i},\bar{j},x}} &= \underbrace{(H^x h)^{\delta'_{\bar{i},\bar{j},x}}}_{\Psi_{1,1}} \cdot (H^x h)^{\sigma'_{\bar{i},\bar{j}} \cdot \sum_{k' \in [l], \rho^*(k') \notin S_{(i,j)}} \frac{b_{k'}}{x - \rho^*(k')}} \cdot (H^x h)^{\sum_{k' \in [l], \rho^*(k') \notin S_{(i,j)}} \sum_{t' \in [n]} \frac{\bar{u}_{t'} b_{k'} a^{q+1-t'}}{x - \rho^*(k')}} \\
&= \Psi_{1,1} \cdot \Big(\prod_{\substack{k' \in [l] \\ \rho^*(k') \notin S_{(i,j)}}} \big(g^{\eta x + \beta} \prod_{k \in [l]} \prod_{t \in [n]} (g^{a^t / b_k^2})^{(x - \rho^*(k)) A^*_{k,t}} \big)^{\frac{\sigma'_{\bar{i},\bar{j}} b_{k'}}{x - \rho^*(k')}} \Big) \\
&\quad \cdot \Big(\prod_{\substack{k' \in [l] \\ \rho^*(k') \notin S_{(i,j)}}} \prod_{t' \in [n]} \big(g^{\eta x + \beta} \prod_{k \in [l]} \prod_{t \in [n]} (g^{a^t / b_k^2})^{(x - \rho^*(k)) A^*_{k,t}} \big)^{\frac{\bar{u}_{t'} b_{k'} a^{q+1-t'}}{x - \rho^*(k')}} \Big)
\end{aligned}$$

$$= \Psi_{1,1} \cdot \Big(\underbrace{\prod_{\substack{k' \in [l] \\ \rho^*(k') \notin S_{(i,j)}}} (g^{b_{k'}})^{\sigma'_{\bar{i},\bar{j}} \cdot (\eta x + \beta)/(x - \rho^*(k'))}}_{\Psi_{1,2}} \Big) \cdot \Big(\prod_{k \in [l]} \prod_{t \in [n]} \prod_{\substack{k' \in [l] \\ \rho^*(k') \notin S_{(i,j)}}} (g^{a^t b_{k'}/b_k^2})^{\sigma'_{\bar{i},\bar{j}} A^*_{k,t} \frac{x - \rho^*(k)}{x - \rho^*(k')}} \Big)$$

$$\cdot \underbrace{\Big(\prod_{\substack{k' \in [l] \\ \rho^*(k') \notin S_{(i,j)}}} \prod_{t' \in [n]} (g^{b_{k'} a^{q+1-t'}})^{\bar{u}_{t'}(\eta x + \beta)/(x - \rho^*(k'))} \Big)}_{\Psi_{1,3}}$$

$$\cdot \Big(\prod_{k \in [l]} \prod_{t \in [n]} \prod_{\substack{k' \in [l] \\ \rho^*(k') \notin S_{(i,j)}}} \prod_{t' \in [n]} (g^{a^{q+1-t'+t} b_{k'}/b_k^2})^{A^*_{k,t} \bar{u}_{t'} \frac{x - \rho^*(k)}{x - \rho^*(k')}} \Big)$$

$$= \Psi_{1,1} \cdot \Psi_{1,2} \cdot \underbrace{\Big(\prod_{\substack{k \in [l] \\ \rho(k) \in S_{(i,j)}}} \prod_{t \in [n]} \prod_{\substack{k' \in [l] \\ \rho^*(k') \notin S_{(i,j)}}} (g^{a^t b_{k'}/b_k^2})^{\sigma'_{\bar{i},\bar{j}} A^*_{k,t} \frac{x - \rho^*(k)}{x - \rho^*(k')}} \Big)}_{\Psi_{1,4} \quad (for\ \rho(k) \in S_{(i,j)})}$$

$$\cdot \underbrace{\Big(\prod_{\substack{k \in [l] \\ \rho(k) \notin S_{(i,j)}}} \prod_{t \in [n]} \prod_{\substack{k' \in [l] \setminus \{k\} \\ \rho^*(k') \notin S_{(i,j)}}} (g^{a^t b_{k'}/b_k^2})^{\sigma'_{\bar{i},\bar{j}} A^*_{k,t} \frac{x - \rho^*(k)}{x - \rho^*(k')}} \Big)}_{\Psi_{1,5} \quad (for\ \rho(k) \notin S_{(i,j)}, k' \neq k)} \cdot \underbrace{\Big(\prod_{\substack{k \in [l] \\ \rho(k) \notin S_{(i,j)}}} \prod_{t \in [n]} (g^{a^t/b_k})^{\sigma'_{\bar{i},\bar{j}} A^*_{k,t}} \Big)}_{\Psi_{1,6} \quad (for\ \rho(k) \notin S_{(i,j)}, k' = k)} \cdot \Psi_{1,3}$$

$$\cdot \underbrace{\Big(\prod_{\substack{k \in [l] \\ \rho(k) \in S_{(i,j)}}} \prod_{t \in [n]} \prod_{\substack{k' \in [l] \\ \rho^*(k') \notin S_{(i,j)}}} \prod_{t' \in [n]} (g^{a^{q+1-t'+t} b_{k'}/b_k^2})^{A^*_{k,t} \bar{u}_{t'} \frac{x - \rho^*(k)}{x - \rho^*(k')}} \Big)}_{\Psi_{1,7} \quad (for\ \rho(k) \in S_{(i,j)})}$$

$$\cdot \underbrace{\Big(\prod_{\substack{k \in [l] \\ \rho(k) \notin S_{(i,j)}}} \prod_{t \in [n]} \prod_{\substack{k' \in [l] \setminus \{k\} \\ \rho^*(k') \notin S_{(i,j)}}} \prod_{t' \in [n]} (g^{a^{q+1-t'+t} b_{k'}/b_k^2})^{A^*_{k,t} \bar{u}_{t'} \frac{x - \rho^*(k)}{x - \rho^*(k')}} \Big)}_{\Psi_{1,8} \quad (for\ \rho(k) \notin S_{(i,j)}, k' \neq k)}$$

$$\cdot \underbrace{\Big(\prod_{\substack{k \in [l] \\ \rho(k) \notin S_{(i,j)}}} \prod_{t \in [n]} \prod_{t' \in [n]} (g^{a^{q+1-t'+t}/b_k})^{A^*_{k,t} \bar{u}_{t'}} \Big)}_{(for\ \rho(k) \notin S_{(i,j)}, k' = k)}$$

$$= \Psi_1 \cdot \Big(\prod_{\substack{k \in [l] \\ \rho(k) \notin S_{(i,j)}}} \prod_{t \in [n]} \prod_{t' \in [n]} (g^{a^{q+1-t'+t}/b_k})^{A^*_{k,t} \bar{u}_{t'}} \Big),$$

$$G^{-\sigma_{\bar{i},\bar{j}}} = G^{-\sigma'_{\bar{i},\bar{j}}} \big(g^{\theta} \prod_{k \in [l]} \prod_{t \in [n]} (g^{a^t/b_k})^{A^*_{k,t}} \big)^{-\sum_{t' \in [n]} \bar{u}_{t'} a^{q+1-t'}}$$

$$= \underbrace{G^{-\sigma'_{\bar{i},\bar{j}}\left(\prod_{t'\in[n]}(g^{a^{q+1-t'}})^{-\theta\bar{u}_{t'}}\right)}}_{\Psi_2} \cdot \left(\prod_{k\in[l]}\prod_{t\in[n]}\prod_{t'\in[n]}(g^{a^{q+1-t'+t}/b_k})^{-A^*_{k,t}\bar{u}_{t'}}\right)$$
$$= \Psi_2 \cdot \left(\prod_{k\in[l]}\prod_{t\in[n]}\prod_{t'\in[n]}(g^{a^{q+1-t'+t}/b_k})^{-A^*_{k,t}\bar{u}_{t'}}\right),$$

where $\Psi_1 = \Psi_{1,1}\cdot\Psi_{1,2}\cdot\cdots\cdot\Psi_{1,8}$ and Ψ_2 can be calculated using the suitable terms of the assumption.

D Proof of Our Modified Source Group Parallel BDHE Assumption

In this section, we give a lower bound to the complexity of our modified source group parallel BDHE assumption. The proof is similar to that of the Source Group q-Parallel BDHE Assumption [19], which is given in [19, Appendix B] in the generic group model. In the generic group model [29], an adversary does not have direct access to the group. It must interact with an oracle to perform the group operation and obtain "handles" for new elements. Also, it can only use handles previously received from the oracle. We consider an experiment where an adversary is given handles for the group elements given out in the assumption as well as a handle for the challenge term T_β (here, β is a uniformly random bit). The adversary may interact with the oracle to perform group operations and pairings, and gets handles in return as the results from these operations. Finally, the adversary guesses the bit β. The difference between the adversary's success probability and one half is defined as its advantage. We refer readers to [3,13] for other examples of using the generic group model for justifying assumptions in bilinear groups. We denote $a, c, d, b_1, \ldots, b_q$ as variables over $\mathbb{Z}_p$, and define $\mathcal{M}$ as the following set of rational functions over these variables:

$$\begin{array}{ll}
\mathcal{M} := \{\ 1, d, cd, da^q, & \\
a^i,\ b_j,\ a^ib_j,\ a^i/b_j^2,\ cdb_j & \forall i,j\in[q], \\
a^i/b_j & \forall i\in[2q]\setminus\{q+1\}, j\in[q], \\
a^ib_{j'}/b_j^2 & \forall i\in[2q], j,j'\in[q]\ s.t.\ j'\neq j, \\
cda^ib_{j'}/b_j, cda^ib_{j'}/b_j^2 & \forall i\in[q], j,j'\in[q]\ s.t.\ j\neq j'\ \}
\end{array}$$

These are the exponents of the group elements given in our modified source group q-parallel BDHE assumption. Let $E(\mathcal{M})$ be the set of all pairwise products of functions in $\mathcal{M}$. It represents the exponents of elements in $\mathbb{G}_T$ that can be obtained by pairing elements with exponents in $\mathcal{M}$. We say a function T is *dependent* on a set of functions $\mathcal{S} = \{S_1, \ldots, S_k\}$ if there exist constants $r_1, \ldots, r_k \in \mathbb{Z}_p$ such that $T = r_1S_1 + \cdots + r_kS_k$. This is an equality of functions over $\mathbb{Z}_p$, and hence hold for *all* settings of the variables. If no such constants exist, we say that T is *independent* of $\mathcal{S}$.

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

Proof. As every element in $\mathcal{M}\cup\{ca^{q+1}\}$ and $E(\mathcal{M})\cup ca^{q+1}(\mathcal{M}\setminus M)$ is a ratio of monomials, the only way that $M\cdot ca^{q+1}$ can be dependent on $E(\mathcal{M})\cup ca^{q+1}(\mathcal{M}\setminus M)$ is if it is *contained* in $E(\mathcal{M})\cup ca^{q+1}(\mathcal{M}\setminus M)$. First, c^2a^{2q+2} is not in $E(\mathcal{M})\cup ca^{q+1}\mathcal{M}$, and for any $M\in\mathcal{M}$, $ca^{q+1}M \notin ca^{q+1}(\mathcal{M}\setminus M)$. Thus it suffices to show that for any M, $ca^{q+1}M \notin E(\mathcal{M})$. In other words, we show

that $E(\mathcal{M})$ does not intersect with the set $ca^{q+1}\mathcal{M}$, which is formed by multiplying each element of $\mathcal{M}$ by ca^{q+1}. To see this, we examine the set $ca^{q+1}\mathcal{M}$. By definition, we have that

$$\begin{aligned} ca^{q+1}\mathcal{M} = \big\{ & ca^{q+1}, cda^{q+1}, c^2da^{q+1}, cda^{2q+1}, \\ & ca^{q+1+i}, ca^{q+1}b_j, ca^{q+1+i}b_j, ca^{q+1+i}/b_j^2, c^2da^{q+1}b_j \qquad \forall i,j \in [q], \\ & ca^{q+1+i}/b_j \qquad \forall i \in [2q] \setminus \{q+1\}, j \in [q], \\ & ca^{q+1+i}b_{j'}/b_j^2 \qquad \forall i \in [2q], j, j' \in [q] \ s.t. \ j' \neq j, \\ & c^2da^{q+1+i}b_{j'}/b_j, c^2da^{q+1+i}b_{j'}/b_j^2 \qquad \forall i \in [q], j, j' \in [q] \ s.t. \ j \neq j' \ \big\}. \end{aligned}$$

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

We now proceed similarly to the proof strategy in [3,13,19] to establish the following theorem:

Theorem 5. *For any adversary $\mathcal{A}$ that makes Q queries to the oracles computing the group operations in $\mathbb{G}, \mathbb{G}_T$ and the bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, the advantages of $\mathcal{A}$ against the modified source group q-parallel BDHE assumption in the generic group model is at most $O(\frac{Q^2q}{p})$.*

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

E Revocable KP-ABE and Blackbox Traceability

In this section, we define Revocable KP-ABE (or R-KP-ABE for short) and its security, which are based on conventional (non-traceable, non-revocable) KP-ABE (e.g. [11,17,26]). Similar to the traceable CP-ABE in [20], in our 'functional' definition, we explicitly assign and identify users using unique indices. Then we formalize traceability against attributes-specific decryption blackbox on R-KP-ABE.

E.1 Revocable KP-ABE

A Revocable Key-Policy Attribute-Based Encryption (R-KP-ABE) scheme consists of four algorithms:

$\mathsf{Setup}(\lambda, N) \rightarrow (\mathsf{PP}, \mathsf{MSK})$. The algorithm takes as input a security parameter λ and the number of users in the system N, runs in polynomial time in λ, and outputs a public parameter PP and a master secret key MSK. We assume that PP contains the description of the attribute universe $\mathcal{U}$[5].

$\mathsf{KeyGen}(\mathsf{PP}, \mathsf{MSK}, \mathbb{A}) \rightarrow \mathsf{SK}_{k,\mathbb{A}}$. The algorithm takes as input the public parameter PP, the master secret key MSK, and an access policy $\mathbb{A}$ over $\mathcal{U}$, and outputs a private decryption key $\mathsf{SK}_{k,\mathbb{A}}$, which is assigned and identified by a unique index $k \in [N]$.

$\mathsf{Encrypt}(\mathsf{PP}, M, R, S) \rightarrow CT_{R,S}$. The algorithm takes as input PP, a message M, a revocation list $R \subseteq [N]$, and an attribute set $S \subseteq \mathcal{U}$, and outputs a ciphertext $CT_{R,S}$. (R, S) is included in $CT_{R,S}$.

$\mathsf{Decrypt}(\mathsf{PP}, CT_{R,S}, \mathsf{SK}_{k,\mathbb{A}}) \rightarrow M$ or $\perp$. The algorithm takes as input PP, a ciphertext $CT_{R,S}$, and a decryption key $\mathsf{SK}_{k,\mathbb{A}}$. If $(k \in [N] \setminus R)$ AND $(S\ satisfies\ \mathbb{A})$, the algorithm outputs a message M, otherwise it outputs $\perp$ indicating the failure of decryption.

Correctness. The scheme must satisfy the following correctness property: For any access policy $\mathbb{A}$ over $\mathcal{U}$, $k \in [N]$, revocation list $R \subseteq [N]$, attribute set $S \subseteq \mathcal{U}$, and message M, suppose $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(\lambda, N)$, $\mathsf{SK}_{k,\mathbb{A}} \leftarrow \mathsf{KeyGen}(\mathsf{PP}, \mathsf{MSK}, \mathbb{A})$, $CT_{R,S} \leftarrow \mathsf{Encrypt}(\mathsf{PP}, M, R, S)$. If $(k \in [N] \setminus R) \wedge (S$ satisfies $\mathbb{A})$ then $\mathsf{Decrypt}(\mathsf{PP}, CT_{R,S}, \mathsf{SK}_{k,\mathbb{A}}) = M$.

Security. The security of the R-KP-ABE is defined by the following game.

$\mathsf{Game}_{\mathsf{MH}}$. This message-hiding game is defined between a challenger and an adversary $\mathcal{A}$.

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

Phase 1. For $i = 1$ to Q_1, $\mathcal{A}$ adaptively submits (index, access policy) pair $(k_i, \mathbb{A}_{k_i})$. The challenger responds with $\mathsf{SK}_{k_i,\mathbb{A}_{k_i}}$.

Challenge. $\mathcal{A}$ submits two equal-length messages M_0, M_1 and a (revocation list, attribute set) pair (R^*, S^*). The challenger flips a random coin $b \in \{0, 1\}$, and sends $CT_{R^*,S^*} \leftarrow \mathsf{Encrypt}(\mathsf{PP}, M_b, R^*, S^*)$ to $\mathcal{A}$.

Phase 2. For $i = Q_1 + 1$ to Q, $\mathcal{A}$ adaptively submits (index, access policy) pair $(k_i, \mathbb{A}_{k_i})$. The challenger responds with $\mathsf{SK}_{k_i,\mathbb{A}_{k_i}}$.

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

$\mathcal{A}$ wins the game if $b' = b$ under the **restriction** that none of the queried $\{(k_i, \mathbb{A}_{k_i})\}_{i=1}^{Q}$ can satisfy $(k_i \in [N] \setminus R^*)$ AND $(S^*\ satisfies\ \mathbb{A}_{k_i})$. The advantage of $\mathcal{A}$ is defined as $\mathsf{MHAdv}_{\mathcal{A}} = |\Pr[b' = b] - \frac{1}{2}|$.

Definition 6. *An N-user R-KP-ABE scheme is secure if for all probabilistic polynomial time (PPT) adversaries $\mathcal{A}$, $\mathsf{MHAdv}_{\mathcal{A}}$ is negligible in λ.*

We say that an N-user R-KP-ABE scheme is *selectively* secure if we add an **Init** stage before **Setup** where the adversary commits to the challenge attribute set S^*.

It is worth noticing that: (1) although the KeyGen algorithm is responsible for determining/assigning the index of each user's decryption key, to capture the security that an adversary can adaptively choose decryption keys to corrupt, the above model allows $\mathcal{A}$ to specify the index when querying for a key, i.e., for $i = 1$ to Q, $\mathcal{A}$ submits pairs of $(k_i, \mathbb{A}_{k_i})$ for decryption keys with access policies corresponding to $\mathbb{A}_{k_i}$, where $Q \leq N$, $k_i \in [N]$, and $k_i \neq k_j$ $\forall 1 \leq i \neq j \leq Q$ (this is to guarantee that each user/key can be *uniquely* identified by an index); and (2) for $k_i \neq k_j$ we do not require $\mathbb{A}_{k_i} \neq \mathbb{A}_{k_j}$, i.e., different users/keys may have the same access policy.

[5] For large universe and also in our work, the attribute universe depends only on the size of the underlying group $\mathbb{G}$, which depends on λ and the group generation algorithm.

Remark: (1) The R-KP-ABE defined above extends the conventional definition for non-revocable KP-ABE [11,17,26], where the revocation list R is always empty. (2) For traceability, we explicitly assign a unique index to each user's decryption key. Predefining the number of users N in the system is indeed a weakness but is also a necessary price to pay for achieving blackbox traceability, but we stress that in practice, this should not incur any noticeable concern, and in fact, all the existing blackbox traceable systems (e.g. [6,7,9,14,20]) have the same setting. (3) When the revocation list R needs an update due to, for example, some decryption keys being compromised or some users leaving the system, the updated R needs to be disseminated to encrypting parties. In practice, this can be done in a similar way to the certificate revocation list distribution in the existing Public Key Infrastructure, namely an authority may update R, and publish it together with the authority's signature generated on it. There are many ways for the encrypting parties to obtain a copy of the updated R, for example, via RSS feeds. (4) From the view of the public, R is just a set of numbers (in $[N]$). These numbers (or indices) do not have to provide any information on the corresponding users, in fact, besides the authority who runs KeyGen, each user only knows his/her own index. Also, encrypting parties do not need to know the indices of any users in order to encrypt but only the attribute sets. Although associating a revocation list with a ciphertext might make the resulting KP-ABE look less purely attribute-based, it does not undermine the capability of KP-ABE, that is, enabling fine-grained access control on encrypted messages.

E.2 Blackbox Traceability

An attributes-specific decryption blackbox $\mathcal{D}$ in the setting of R-KP-ABE is viewed as a probabilistic circuit that can decrypt ciphertexts generated under some specific pair of revocation list and attribute set. *In particular, an attributes-specific decryption blackbox $\mathcal{D}$ is described by a (revocation list, attribute set) pair $(R_{\mathcal{D}}, S_{\mathcal{D}})$ and a non-negligible probability value ϵ (i.e. $\epsilon = 1/f(\lambda)$ for some polynomial f), and this blackbox $\mathcal{D}$ can decrypt ciphertexts generated under $(R_{\mathcal{D}}, S_{\mathcal{D}})$ with probability at least ϵ.* Such a blackbox can reflect most practical scenarios, which include the key-like decryption blackbox for sale where an explicit description of the blackbox's decryption ability is given and decryption blackbox "found in the wild" where only some clue on the attribute set of the ciphertext that the blackbox can decrypt may be found, similar to that discussed in [20]. In particular, once a blackbox is found being able to decrypt ciphertexts (regardless of how this is found, for example, an explicit description of the blackbox's decryption ability is given, or the law enforcement agency finds some clue), we can regard it as an attributes-specific decryption blackbox with the corresponding (revocation list, attribute set) pair (which is associated to the ciphertext). And for a decryption blackbox, if multiple (revocation list, attribute set) pairs are found that corresponding ciphertexts can be decrypted by it with non-negligible probability, we can regard the blackbox as multiple attributes-specific decryption blackboxes, each with a different (revocation list, attribute set) pair.

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

$\mathsf{Trace}^{\mathcal{D}}(\mathsf{PP}, R_{\mathcal{D}}, S_{\mathcal{D}}, \epsilon) \rightarrow \mathbb{K}_T \subseteq [N]$. Trace *is an oracle algorithm that interacts with an attributes-specific decryption blackbox $\mathcal{D}$. By given the public parameter PP, a revocation list $R_{\mathcal{D}}$, an attribute set $S_{\mathcal{D}}$, and a probability value ϵ, the algorithm runs in time polynomial in λ and $1/\epsilon$, and outputs an index set $\mathbb{K}_T \subseteq [N]$ which identifies the set of malicious users. Note that ϵ has to be polynomially related to λ, i.e. $\epsilon = 1/f(\lambda)$ for some polynomial f.*

The following tracing game captures the notion of fully collusion-resistant traceability against attributes-specific decryption blackbox. In the game, the adversary targets to build a decryption blackbox $\mathcal{D}$ that can decrypt ciphertexts under some (revocation list, attribute set) pair $(R_{\mathcal{D}}, S_{\mathcal{D}})$. The tracing algorithm, on the other side, is designed to extract the index of at least one of the malicious users whose decryption keys have been used for constructing $\mathcal{D}$.

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

Setup. The challenger runs $\mathsf{Setup}(\lambda, N)$ and gives the public parameter PP to $\mathcal{A}$.
Key Query. For $i = 1$ to Q, $\mathcal{A}$ adaptively submits (index, access policy) pair $(k_i, \mathbb{A}_{k_i})$. The challenger responds with $\mathsf{SK}_{k_i,\mathbb{A}_{k_i}}$.
Decryption Blackbox Generation. $\mathcal{A}$ outputs a decryption blackbox $\mathcal{D}$ associated with a (revocation list, attribute set) pair $(R_{\mathcal{D}}, S_{\mathcal{D}})$ and a non-negligible probability value ϵ.
Tracing. The challenger runs $\mathsf{Trace}^{\mathcal{D}}(\mathsf{PP}, R_{\mathcal{D}}, S_{\mathcal{D}}, \epsilon)$ to obtain an index set $\mathbb{K}_T \subseteq [N]$.

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

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

We denote by $\mathsf{TRAdv}_{\mathcal{A}}$ the probability that $\mathcal{A}$ wins.

For a useful attributes-specific decryption blackbox $\mathcal{D}$, the traced $\mathbb{K}_T$ must satisfy $(\mathbb{K}_T \neq \emptyset) \wedge (\mathbb{K}_T \subseteq \mathbb{K}_{\mathcal{D}}) \wedge (\exists k_t \in \mathbb{K}_T\ s.t.\ (k_t \in [N] \backslash R_{\mathcal{D}})$ AND $(S_{\mathcal{D}}\ satisfies\ \mathbb{A}_{k_t}))$. (1) $(\mathbb{K}_T \neq \emptyset) \wedge (\mathbb{K}_T \subseteq \mathbb{K}_{\mathcal{D}})$ captures the preliminary traceability that the tracing algorithm can extract at least one malicious user and the coalition of malicious users cannot frame any innocent user. (2) $(\exists k_t \in \mathbb{K}_T\ s.t.\ (k_t \in [N] \setminus R_{\mathcal{D}})$ AND $(S_{\mathcal{D}}\ satisfies\ \mathbb{A}_{k_t}))$ captures the *strong traceability* that the tracing algorithm can extract at least one malicious user whose decryption key enables $\mathcal{D}$ to have the decryption ability corresponding to $(R_{\mathcal{D}}, S_{\mathcal{D}})$, i.e. whose index is not in $R_{\mathcal{D}}$ and whose access policy is satisfied by $S_{\mathcal{D}}$. Strong traceability is desirable in practice, since it can defend against attacks where colluding traitors may build $\mathcal{D}$ in a smart manner so that $\mathcal{D}$ will be traced to only a user whose index is in $R_{\mathcal{D}}$ or whose access policy is not satisfied by $S_{\mathcal{D}}$, which should not happen for a secure R-KP-ABE. We refer to [14,20] on why strong traceability is desirable. Note that, as of [6,7,9,14,20], we are modeling a stateless (resettable) decryption blackbox – such a blackbox is just an oracle and maintains no state between activations. Also note that we are modeling public traceability, namely, the Trace algorithm does not need any secrets and anyone can perform the tracing from the public parameter only.

Definition 7. *An N-user R-KP-ABE scheme is traceable against attributes-specific decryption blackbox if for all PPT adversaries $\mathcal{A}$, $\mathsf{TRAdv}_{\mathcal{A}}$ is negligible in λ.*

We say that an N-user R-KP-ABE is *selectively* traceable against attributes-specific decryption blackbox if we add an **Init** stage before **Setup** where the adversary commits to the access policy $\mathbb{A}_{\mathcal{D}}$.

In the traceable R-KP-ABE above, it is possible to trace *all the active decryption keys* in the blackbox. In particular, given a decryption blackbox $\mathcal{D}$ described by $(R_{\mathcal{D}}, S_{\mathcal{D}})$ and non-negligible probability ϵ, we can run Trace to obtain an index set $\mathbb{K}_T$ so that $(\mathbb{K}_T \neq \emptyset) \wedge (\mathbb{K}_T \subseteq \mathbb{K}_{\mathcal{D}}) \wedge$

$(\exists k_t \in \mathbb{K}_T \; s.t. \; (k_t \in [N] \setminus R_{\mathcal{D}}) \; \mathsf{AND} \; (S_{\mathcal{D}} \; satisfies \; \mathbb{A}_{k_t}))$. Then, we can set a new revocation list $R'_{\mathcal{D}} = R_{\mathcal{D}} \cup \{k_t \in \mathbb{K}_T \mid (k_t \in [N] \setminus R_{\mathcal{D}}) \; \mathsf{AND} \; (S_{\mathcal{D}} \; satisfies \; \mathbb{A}_{k_t})\}$ and test whether $\mathcal{D}$ can decrypt ciphertexts under $(R'_{\mathcal{D}}, S_{\mathcal{D}})$. If $\mathcal{D}$ can still decrypt the ciphertexts with non-negligible probability ϵ', we can run Trace on $(R'_{\mathcal{D}}, S_{\mathcal{D}}, \epsilon')$ and obtain a new index set $\mathbb{K}'_T$, where $(\mathbb{K}'_T \neq \emptyset) \wedge (\mathbb{K}'_T \subseteq \mathbb{K}_{\mathcal{D}}) \wedge (\exists k_t \in \mathbb{K}'_T \; s.t. \; (k_t \in [N] \setminus R'_{\mathcal{D}}) \; \mathsf{AND} \; (S_{\mathcal{D}} \; satisfies \; \mathbb{A}_{k_t}))$. By repeating this process, iteratively expanding the revocation list, until $\mathcal{D}$ can no longer decrypt the corresponding ciphertexts, we have finished finding out *all the active* malicious users of $\mathcal{D}$.

F Augmented R-KP-ABE

We now define Augmented R-KP-ABE (or AugR-KP-ABE for short) from the R-KP-ABE above, formalize its security notions, then show that a secure AugR-KP-ABE can be transformed to a R-KP-ABE with blackbox traceability. In Appendix G, we propose a concrete construction of AugR-KP-ABE.

F.1 Definitions

An AugR-KP-ABE scheme has four algorithms: $\mathsf{Setup}_\mathsf{A}$, $\mathsf{KeyGen}_\mathsf{A}$, $\mathsf{Encrypt}_\mathsf{A}$, and $\mathsf{Decrypt}_\mathsf{A}$. The setup and key generation algorithms are the same as that of R-KP-ABE. For the encryption algorithm, it takes one more parameter $\bar{k} \in [N+1]$ as input, and is defined as follows.

$\mathsf{Encrypt}_\mathsf{A}(\mathsf{PP}, M, R, S, \bar{k}) \to CT_{R,S}$. The algorithm takes as input PP, a message M, a revocation list $R \subseteq [N]$, an attribute set S, and an index $\bar{k} \in [N+1]$, and outputs a ciphertext $CT_{R,S}$. (R, S) **is included in** $CT_{R,S}$**, but the value of** $\bar{k}$ **is not.**

The decryption algorithm is also defined in the same way as that of R-KP-ABE. However, the correctness definition is changed to the following.

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

Note that during decryption, as long as $(k \in [N] \setminus R) \wedge (S \text{ satisfies } \mathbb{A})$, the decryption algorithm outputs a message, but only when $k \geq \bar{k}$, the output message is equal to the correct message, that is, if and only if $(k \in [N] \setminus R) \wedge (S \text{ satisfies } \mathbb{A}) \wedge (k \geq \bar{k})$, can $\mathsf{SK}_{k,\mathbb{A}}$ correctly decrypt a ciphertext under $(R, S, \bar{k})$. If we always set $\bar{k} = 1$, the functions of AugR-KP-ABE are identical to that of R-KP-ABE. In fact, the idea behind transforming an AugR-KP-ABE to a blackbox traceable R-KP-ABE, that we will show shortly, is to construct an AugR-KP-ABE with index-hiding property, and then always sets $\bar{k} = 1$ in normal encryption, while using $\bar{k} \in [N+1]$ to generate ciphertexts for tracing.

Security. We define the security of AugR-KP-ABE in two games. The first game is a **message-hiding game** and says that a ciphertext created using index $N+1$ is unreadable by anyone. The second game is an **index-hiding game** and captures the intuition that a ciphertext created using index $\bar{k}$ reveals no non-trivial information about $\bar{k}$.

$\mathsf{Game}^\mathsf{A}_\mathsf{MH}$. The **message-hiding game** $\mathsf{Game}^\mathsf{A}_\mathsf{MH}$ is similar to $\mathsf{Game}_\mathsf{MH}$ except that during the **Challenge** phase, the challenge ciphertext is computed as $CT_{R^*,S^*} \leftarrow \mathsf{Encrypt}_\mathsf{A}(\mathsf{PP}, M_b, R^*, S^*, N+1)$,

and the original **restriction** in $\mathsf{Game}_{\mathsf{MH}}$ no longer applies in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}}$. In particular, the **message-hiding game** $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}}$ proceeds as follows:

Setup. The challenger runs $\mathsf{Setup}_{\mathsf{A}}(\lambda, N)$ and gives the public parameter PP to $\mathcal{A}$.
Phase 1. For $i = 1$ to Q_1, $\mathcal{A}$ adaptively submits (index, access policy) pair $(k_i, \mathbb{A}_{k_i})$. The challenger responds with $\mathsf{SK}_{k_i,\mathbb{A}_{k_i}}$.
Challenge. $\mathcal{A}$ submits two equal-length messages M_0, M_1 and a (revocation list, attribute set) pair (R^*, S^*). The challenger flips a random coin $b \in \{0,1\}$, and sends $CT_{R^*,S^*} \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M_b, R^*, S^*, N+1)$ to $\mathcal{A}$.
Phase 2. For $i = Q_1 + 1$ to Q, $\mathcal{A}$ adaptively submits (index, access policy) pair $(k_i, \mathbb{A}_{k_i})$. The challenger responds with $\mathsf{SK}_{k_i,\mathbb{A}_{k_i}}$.
Guess. $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ for b.

$\mathcal{A}$ wins the game if $b' = b$. The advantage of the adversary $\mathcal{A}$ in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}}$ is defined as $\mathsf{MH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}} = |\Pr[b' = b] - \frac{1}{2}|$.

Definition 8. *An N-user Augmented R-KP-ABE scheme is message-hiding in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}}$ if for all PPT adversaries $\mathcal{A}$ the advantage $\mathsf{MH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}$ is negligible in λ.*

$\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$. In the **index-hiding game**, we require that, for any (revocation list, attribute set) pair (R^*, S^*), an adversary cannot distinguish between a ciphertext under $(R^*, S^*, \bar{k})$ and $(R^*, S^*, \bar{k}+1)$ without a decryption key $\mathsf{SK}_{\bar{k},\mathbb{A}_{\bar{k}}}$, where $(\bar{k} \in [N] \setminus R^*) \wedge (S^* \; satisfies \; \mathbb{A}_{\bar{k}})$. 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 $\mathsf{Setup}_{\mathsf{A}}(\lambda, N)$ and gives the public parameter PP to $\mathcal{A}$.
Phase 1. For $i = 1$ to Q_1, $\mathcal{A}$ adaptively submits (index, access policy) pair $(k_i, \mathbb{A}_{k_i})$. The challenger responds with $\mathsf{SK}_{k_i,\mathbb{A}_{k_i}}$.
Challenge. $\mathcal{A}$ submits a message M and a (revocation list, attribute set) pair (R^*, S^*). The challenger flips a random coin $b \in \{0,1\}$, and sends $CT_{R^*,S^*} \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, R^*, S^*, \bar{k}+b)$ to $\mathcal{A}$.
Phase 2. For $i = Q_1 + 1$ to Q, $\mathcal{A}$ adaptively submits (index, access policy) pair $(k_i, \mathbb{A}_{k_i})$. The challenger responds with $\mathsf{SK}_{k_i,\mathbb{A}_{k_i}}$.
Guess. $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ for b.

$\mathcal{A}$ wins the game if $b' = b$ under the **restriction** that none of the queried pairs $\{(k_i, \mathbb{A}_{k_i})\}_{i=1}^{Q}$ can satisfy $(k_i = \bar{k}) \wedge (k_i \in [N] \setminus R^*) \wedge (S^* \; satisfies \; \mathbb{A}_{k_i})$. 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 9. *An N-user Augmented R-KP-ABE scheme 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.*

We say that an Augmented R-KP-ABE scheme is *selectively* index-hiding if we add an **Init** stage before **Setup** where the adversary commits to the challenge attribute set S^*.

F.2 The Reduction of Traceable R-KP-ABE to Augmented R-KP-ABE

Let $\Sigma_{\mathsf{A}} = (\mathsf{Setup}_{\mathsf{A}}, \mathsf{KeyGen}_{\mathsf{A}}, \mathsf{Encrypt}_{\mathsf{A}}, \mathsf{Decrypt}_{\mathsf{A}})$ be an AugR-KP-ABE, define $\mathsf{Encrypt}(\mathsf{PP}, M, R, S) = \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, R, S, 1)$, then $\Sigma = (\mathsf{Setup}_{\mathsf{A}}, \mathsf{KeyGen}_{\mathsf{A}}, \mathsf{Encrypt}, \mathsf{Decrypt}_{\mathsf{A}})$ is a R-KP-ABE derived from Σ_{A}. In the following, we show that if Σ_{A} is message-hiding and index-hiding, then Σ is secure (w.r.t. Def. 6). Furthermore, we propose a tracing algorithm Trace for Σ and show that if Σ_{A} is message-hiding and index-hiding, then Σ (equipped with Trace) is traceable (w.r.t. Def. 7).

F.2.1 R-KP-ABE Security

Theorem 6. *If Σ_{A} is message-hiding and index-hiding (resp. selectively index-hiding), then Σ is secure (resp. selectively secure).*

Proof. First we need a slightly more elaborate message-hiding game for Σ_{A}. In addition to N, λ, this extended game, denoted as $\mathsf{Game}^{\mathsf{A}}_{\mathsf{EMH}}$, takes as input a parameter $\bar{k} \in [N+1]$ which is only given to the challenger. $\mathsf{Game}^{\mathsf{A}}_{\mathsf{EMH}}$ proceeds as follows:

Setup. The challenger runs $\mathsf{Setup}_{\mathsf{A}}(\lambda, N)$ and gives the public parameter PP to $\mathcal{A}$.
Phase 1. For $i = 1$ to Q_1, $\mathcal{A}$ adaptively submits (index, access policy) pair $(k_i, \mathbb{A}_{k_i})$, and obtains $\mathsf{SK}_{k_i,\mathbb{A}_{k_i}}$.
Challenge. $\mathcal{A}$ submits two equal-length messages M_0, M_1 and a (revocation list, attribute set) pair (R^*, S^*). The challenger flips a random coin $b \in \{0,1\}$, and sends $CT_{R^*,S^*} \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M_b, R^*, S^*, \bar{k})$ to $\mathcal{A}$. This is the only place where $\bar{k}$ is used in the game.
Phase 2. For $i = Q_1 + 1$ to Q, $\mathcal{A}$ adaptively submits (index, access policy) pair $(k_i, \mathbb{A}_{k_i})$, and obtains $\mathsf{SK}_{k_i,\mathbb{A}_{k_i}}$.
Guess. $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ for b.

The adversary $\mathcal{A}$ wins the game if $b' = b$ under the **restriction** that none of the queried pairs $\{(k_i, \mathbb{A}_{k_i})\}_{i=1}^{Q}$ can satisfy $(k_i \in [N] \setminus R^*) \wedge (S^* \; satisfies \; \mathbb{A}_{k_i})$. The advantage of $\mathcal{A}$ is defined as $\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}] = |\Pr[b' = b] - \frac{1}{2}|$.

When $\bar{k} = 1$, the game above, including the **restriction**, is exactly identical to the message-hiding game $\mathsf{Game}_{\mathsf{MH}}$ for Σ, we have $\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[1] = \mathsf{MHAdv}_{\mathcal{A}}$. When $\bar{k} = N+1$, we have that $\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[N+1] \leq \mathsf{MH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}$, since $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}}$ is identical to $\mathsf{Game}^{\mathsf{A}}_{\mathsf{EMH}}$ for $\bar{k} = N+1$, but there is no restriction in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}}$. In the following proof sketch, we will make use of the facts that Σ_{A} is message-hiding and index-hiding to show that $\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[1]$ is negligible, which implies that $\mathsf{MHAdv}_{\mathcal{A}}$ is negligible (i.e. Σ is secure w.r.t. Def. 6).

Suppose that Σ is not secure, i.e. $\mathsf{MHAdv}_{\mathcal{A}} > \epsilon$ for some adversary $\mathcal{A}$ and non-negligible ϵ. $\mathsf{MHAdv}_{\mathcal{A}} > \epsilon$ implies that $\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[1] > \epsilon$. As Σ_{A} is message-hiding, $\mathsf{MH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}$ is negligible (for simplicity, say $\mathsf{MH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}} = 0$), thus $\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[N+1] = 0$. Then, by the standard hybrid argument there exists a $\bar{k} \in [N]$ such that

$$|\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}] - \mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}+1]| > \epsilon/N.$$

In other words, with non-negligible probability, $\mathcal{A}$ is able to distinguish $\mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, R^*, S^*, \bar{k})$ from $\mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, R^*, S^*, \bar{k}+1)$ for some M and (R^*, S^*). But then $\mathcal{A}$ can directly be used to win the index-hiding game $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$.

More specifically, in Appendix H, we show that for any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that for all $\bar{k} = 1, \ldots, N$, we have

$$|\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}] - \mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}+1]| \leq 2 \cdot \mathsf{IH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{B}}[\bar{k}]. \tag{4}$$

Then we have

$$|\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[1] - \mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[N+1]| \leq \sum_{\bar{k}=1}^{N} |\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}] - \mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}+1]| \leq 2\sum_{\bar{k}=1}^{N} \mathsf{IH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{B}}[\bar{k}].$$

But since Σ_{A} is message-hiding and index-hiding, we have that $\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[N+1]$ and $\mathsf{IH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{B}}[\bar{k}]$ for $\bar{k} = 1, \ldots, N$ are negligible for any PPT adversary. Therefore, $\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[1]$ is negligible. The selective case is similar.

F.2.2 R-KP-ABE Traceability

We now propose a tracing algorithm, which uses a general tracing method previously used in [4,22,6,7,9,20], and show that equipped with Trace, Σ is traceable (w.r.t. Def. 7).

$\mathsf{Trace}^{\mathcal{D}}(\mathsf{PP}, R_{\mathcal{D}}, S_{\mathcal{D}}, \epsilon) \rightarrow \mathbb{K}_T \subseteq [N]$: Given an attributes-specific decryption blackbox $\mathcal{D}$ associated with a (revocation list, attribute set) pair $(R_{\mathcal{D}}, S_{\mathcal{D}})$ and probability $\epsilon > 0$, the tracing algorithm works as follows:

1. For $k = 1$ to $N + 1$, do the following:
   (a) Repeat the following $8\lambda(N/\epsilon)^2$ times:
      i. Sample M from the message space at random.
      ii. Let $CT_{R_{\mathcal{D}}, S_{\mathcal{D}}} \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, R_{\mathcal{D}}, S_{\mathcal{D}}, k)$.
      iii. Query oracle $\mathcal{D}$ on input $CT_{R_{\mathcal{D}}, S_{\mathcal{D}}}$, and compare the output of $\mathcal{D}$ with M.
   (b) Let $\hat{p}_k$ be the fraction of times that $\mathcal{D}$ decrypted the ciphertexts correctly.
2. Let $\mathbb{K}_T$ be the set of all $k \in [N]$ for which $\hat{p}_k - \hat{p}_{k+1} \geq \epsilon/(4N)$. Output $\mathbb{K}_T$.

The running time is cubic in N. It can be made (almost) quadratic using binary search instead of a linear scan.

Theorem 7. *If Σ_{A} is message-hiding and index-hiding (resp. selectively index-hiding), then Σ is traceable (resp. selectively traceable).*

Proof. We show that if the blackbox output by the adversary is a useful one then $\mathbb{K}_T$ will satisfy $(\mathbb{K}_T \neq \emptyset) \wedge (\mathbb{K}_T \subseteq \mathbb{K}_{\mathcal{D}}) \wedge (\exists k_t \in \mathbb{K}_T \; s.t. \; (k_t \in [N] \setminus R_{\mathcal{D}}) \wedge (S_{\mathcal{D}} \; satisfies \; \mathbb{A}_{k_t}))$ with overwhelming probability, which implies that the adversary cannot win $\mathsf{Game}_{\mathsf{TR}}$, i.e., $\mathsf{TRAdv}_{\mathcal{A}}$ is negligible. The selective case will be similar. Let $\mathcal{D}$ be the attributes-specific decryption blackbox output by the adversary, and $(R_{\mathcal{D}}, S_{\mathcal{D}})$ be the (revocation list, attribute set) pair describing $\mathcal{D}$. Define

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

where the probability is taken over the random choice of message M and the random coins of $\mathcal{D}$. We have that $p_1 \geq \epsilon$ and p_{N+1} is negligible (for simplicity let $p_{N+1} = 0$). The former follows from the fact that $\mathcal{D}$ is useful, and the latter is because Σ_{A} is message-hiding in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}}$. Then there must exist some $k \in [N]$ such that $p_k - p_{k+1} \geq \epsilon/(2N)$. By the Chernoff bound it follows that with overwhelming probability, $\hat{p}_k - \hat{p}_{k+1} \geq \epsilon/(4N)$. Hence, we have $\mathbb{K}_T \neq \emptyset$.

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

G An Efficient Augmented R-KP-ABE

We propose an AugR-KP-ABE scheme which is highly expressive and efficient with sub-linear overhead in the number of users in the system. It is also *large universe*, where attributes do not need to be enumerated during setup, and the public parameter size is independent of the attribute universe size. We show that this AugR-KP-ABE is message-hiding and selectively index-hiding in the standard model.

Combining this AugR-KP-ABE with the results in Sec. F.2, we obtain a large universe R-KP-ABE which is selectively secure and traceable, and for a fully collusion-resistant blackbox traceable system, the resulting R-KP-ABE is the most efficient one to date, with sub-linear overhead.

To obtain this practical KP-ABE scheme supporting traitor tracing, revocation and large universe, we borrow ideas from the Blackbox Traceable CP-ABE of [20], the Trace and Revoke scheme of [9] and the Large Universe KP-ABE of [26], but the work is not trivial as a straightforward combination of the ideas would result in a scheme which is inefficient, insecure, or is not able to achieve strong traceability, as also discussed in [20]. Specifically, by incorporating the ideas from [9] and [26] into the Augmented CP-ABE of [20], we can obtain a large universe AugR-KP-ABE which is message-hiding, but proving the index-hiding property is a challenging task. The proof techniques for index-hiding in [20] only work if the attribute universe size is polynomial in the security parameter and the parameters of attributes have to be enumerated during setup. They are not applicable to large universe. The proof techniques in [26] are applicable to large universe, but work only for proving security (i.e. message-hiding), while not applicable to index-hiding. To prove index-hiding in the large universe setting, we introduce a new assumption that the index-hiding of our large universe AugR-KP-ABE can be based on. In particular, in the underlying q-2 assumption of [26] on bilinear groups $(p, \mathbb{G}, \mathbb{G}_T, e)$, the challenge term $T \in \mathbb{G}_T$ is $e(g,g)^{abc}$ or a random element, and such a term in the target group could be used to prove the message-hiding as the message space is $\mathbb{G}_T$. To prove the index-hiding, which is based on the ciphertext components in the source group $\mathbb{G}$, we need the challenge term to be in the source group $\mathbb{G}$. Inspired by the Source Group q-Parallel BDHE Assumption in [19], which is a close relative to the (target group) Decisional Parallel BDHE Assumption in [30], we modify the q-2 assumption to its source group version where the challenge term is g^{abc} or a random element in $\mathbb{G}$. Based on this new assumption and with a new crucial proof idea, we prove the index-hiding property for our large universe AugR-KP-ABE. We prove that this new assumption holds in the generic group model.

G.1 Preliminaries

Complexity Assumptions. Besides the Decision 3-Party Diffie-Hellman Assumption (D3DH) and the Decisional Linear Assumption (DLIN) that are used in [9] to achieve traceability in broadcast encryption, the index-hiding property of our AugR-KP-ABE construction will rely on a new assumption, which we refer to as Source Group q-Parallel DBDH Assumption. Here we only review this new assumption, and refer to [9] for the details of the other assumptions.

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

$$
\begin{aligned}
&(p, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G},\\
&g \xleftarrow{R} \mathbb{G}, \quad a, b, c, d, d_1, \ldots, d_q \xleftarrow{R} \mathbb{Z}_p,\\
&D = \big((p, \mathbb{G}, \mathbb{G}_T, e), g, g^a, g^b, g^{cd}, g^d, g^{ad}, g^{(acd)^2},\\
&\qquad g^{d_i},\ g^{acdd_i},\ g^{a^2cdd_i},\ g^{acd/d_i},\ g^{b/d_i^2},\ g^{b^2/d_i^2} \quad \forall i \in [q],\\
&\qquad g^{acdd_i/d_j},\ g^{(acd)^2 d_i/d_j},\ g^{bd_i/d_j^2},\ g^{abcdd_i/d_j^2} \quad \forall i, j \in [q]\ s.t.\ i \neq j\ \big),\\
&T_0 = g^{abc}, T_1 \xleftarrow{R} \mathbb{G}.
\end{aligned}
$$

The advantage of an algorithm $\mathcal{A}$ in breaking the Source Group q-Parallel DBDH Assumption is: $Adv^q_{\mathcal{G},\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(D, T_0) = 1] - \Pr[\mathcal{A}(D, T_1) = 1]|$.

Definition 10. *$\mathcal{G}$ satisfies the Source Group q-Parallel DBDH Assumption if $Adv^q_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ for any PPT algorithm $\mathcal{A}$.*

Note that the q-2 assumption in [26] is a variant of DBDH assumption, where $4q^2 + 2q + 1$ additional input terms are given to the adversary, we refer to it as q-parallel DBDH assumption. The above new assumption is closely related to q-parallel DBDH assumption (i.e. the q-2 assumption in [26]) except that the challenge term g^{abc} remains in the source group, all the input terms replace c with cd, and additional input terms g^d and g^{da} are given to the adversary. The relation between this assumption and the q-2 assumption [26] is analogous to that between the Source Group q-Parallel BDHE Assumption [19] and the Decisional Parallel BDHE Assumption [30], i.e. the challenge term changes from a term in the target group (i.e. $e(g,g)^{abc}$ and $e(g,g)^{ca^{q+1}}$ respectively) to a term in the source group (i.e. g^{abc} and $g^{ca^{q+1}}$ respectively), and the input terms are modified accordingly (i.e. replacing c with cd, and adding g^d). The main difference is that in this new assumption, there is an additional input term g^{da}. Note that giving the term g^{da} does not pose any problem in the generic group model. Intuitively, there are two ways that the adversary may make use of the term g^{da}: (1) pairing g^{da} with the challenge term: since the pairing result of any two input terms would not be $e(g,g)^{a^2bcd}$, the adversary cannot break this new assumption in this way; (2) pairing the challenge term with another input term whose exponent contains d: however, the result could be a random element or one of $\{\ e(g,g)^{abc^2d},\ e(g,g)^{abcd},\ e(g,g)^{a^2bcd},\ e(g,g)^{a^2bc^2dd_i},\ e(g,g)^{a^3bc^2dd_i},\ e(g,g)^{a^2bc^2d/d_i},\ e(g,g)^{a^2bc^2dd_i/d_j},\ e(g,g)^{a^3bc^3d^2d_i/d_j},\ e(g,g)^{a^2b^2c^2dd_i/d_j^2}\ \}$, and as there is no input term which can be paired with g^{da} to obtain any of these terms, the adversary cannot break this new assumption by this way either. In Appendix J, we prove that this assumption holds in the generic group model.

G.2 Augmented R-KP-ABE Construction

Now we propose a large universe Augmented R-KP-ABE, where the attribute universe is $\mathcal{U} = \mathbb{Z}_p$, and we do not need to enumerate all the attributes or their corresponding public parameters during system setup. Note that the notations here are same to that of the Augmented R-CP-ABE construction in Sec. 4.2.

$\mathsf{Setup}_\mathsf{A}(\lambda, N = m^2) \to (\mathsf{PP}, \mathsf{MSK})$. The algorithm calls the group generator $\mathcal{G}(1^\lambda)$ to get $(p, \mathbb{G}, \mathbb{G}_T, e)$, where p is the prime order of $\mathbb{G}$ and $\mathbb{G}_T$ and e is the bilinear map, and sets the attribute universe to $\mathcal{U} = \mathbb{Z}_p$. It then randomly picks:

$$g,\ h,\ f,\ f_1, \ldots, f_m,\ G,\ H\ \in \mathbb{G},\ \ \{\alpha_i,\ r_i,\ z_i \in \mathbb{Z}_p\}_{i\in[m]},\ \{c_j \in \mathbb{Z}_p\}_{j\in[m]},$$

and outputs the public parameter PP and master secret key MSK as

$$\mathsf{PP} = \Big(\ (p, \mathbb{G}, \mathbb{G}_T, e),\ g,\ h,\ f,\ f_1, \ldots, f_m,\ G,\ H,\ \{E_i = e(g,g)^{\alpha_i}, G_i = g^{r_i},\ Z_i = g^{z_i}\}_{i\in[m]},\ \ \{H_j = g^{c_j}\}_{j\in[m]}\ \Big).$$

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

A counter $ctr = 0$ is implicitly included in MSK.

$\mathsf{KeyGen}_\mathsf{A}(\mathsf{PP}, \mathsf{MSK}, (A, \rho)) \to \mathsf{SK}_{(i,j),(A,\rho)}$. (A, ρ) is an LSSS matrix where ρ maps each row A_k of A to an attribute $\rho(k) \in \mathcal{U} = \mathbb{Z}_p$. Let $l \times n$ be the size of A. The algorithm first sets $ctr = ctr+1$ and computes the corresponding index in the form of (i, j) where $1 \le i, j \le m$ and $(i-1)*m+j = ctr$. Then the algorithm randomly chooses $\boldsymbol{u} = (\sigma_{i,j}, u_2, \ldots, u_n) \in \mathbb{Z}_p^n$ and $\{\xi_k \in \mathbb{Z}_p\}_{k\in[l]}$, and outputs a private key $\mathsf{SK}_{(i,j),(A,\rho)} = \big((i,j), (A,\rho),\ K_{i,j}, K'_{i,j}, K''_{i,j}, \{\bar{K}_{i,j,j'}\}_{j'\in[m]\setminus\{j\}}, \{K_{i,j,k,1}, K_{i,j,k,2},$

$K_{i,j,3}\}_{k\in[l]}\big)$ where

$$K_{i,j} = g^{\alpha_i} g^{r_i c_j} (f f_j)^{\sigma_{i,j}},\ K'_{i,j} = g^{\sigma_{i,j}},\ K''_{i,j} = Z_i^{\sigma_{i,j}},\ \ \{\bar{K}_{i,j,j'} = f_{j'}^{\sigma_{i,j}}\}_{j'\in[m]\setminus\{j\}},$$
$$\{K_{i,j,k,1} = f^{(A_k\cdot \boldsymbol{u})} G^{\xi_k},\ K_{i,j,k,2} = (H^{\rho(k)}h)^{-\xi_k},\ K_{i,j,k,3} = g^{\xi_k}\}_{k\in[l]}.$$

$\mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M, R, S, (\bar{i},\bar{j})) \to CT_{R,S}$. $R \subseteq [m,m]$ is a revocation list. $S \subseteq \mathcal{U} = \mathbb{Z}_p$ is an attribute set. This algorithm allows the encrypting party to encrypt a message to the recipients whose (index, access policy) pairs $((i,j),(A,\rho))$ satisfy $\big((i,j)\in[m,m]\setminus R\big) \wedge \big(S\ satisfies\ (A,\rho)\big) \wedge \big((i,j)\geq(\bar{i},\bar{j})\big)$. Let $\bar{R} = [m,m]\setminus R$ and for $i\in[m]$, $\bar{R}_i = \{j' | (i,j')\in\bar{R}\}$, that is, $\bar{R}$ is the non-revoked index list, and $\bar{R}_i$ is the set of non-revoked column index on the i-th row. The algorithm randomly chooses

$$\kappa,\ \tau,\ \ s_1,\ldots,s_m,\ \ t_1,\ldots,t_m\ \in\mathbb{Z}_p,$$
$$\boldsymbol{v}_c,\ \boldsymbol{w}_1,\ldots,\boldsymbol{w}_m\ \in\mathbb{Z}_p^3,$$
$$\delta_x\ \in\mathbb{Z}_p\ \forall x\in S,\ \ \pi\ \in\mathbb{Z}_p.$$

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

$$\boldsymbol{v}_i \in \mathbb{Z}_p^3\ \forall i\in\{1,\ldots,\bar{i}\},$$
$$\boldsymbol{v}_i \in span\{\boldsymbol{\chi}_1,\boldsymbol{\chi}_2\}\ \forall i\in\{\bar{i}+1,\ldots,m\},$$

and creates the ciphertext $\langle R, S,\ (\boldsymbol{R}_i, \boldsymbol{R}'_i, Q_i, Q'_i, Q''_i, T_i)_{i=1}^m, (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^m,\ P, \{P_x, P'_x\}_{x\in S}\rangle$ as follows:

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

$$\boldsymbol{R}_i = g^{\boldsymbol{v}_i},\ \boldsymbol{R}'_i = g^{\kappa\boldsymbol{v}_i},\quad Q_i = g^{s_i},\ Q'_i = (f\prod_{j'\in\bar{R}_i} f_{j'})^{s_i} Z_i^{t_i} f^{\pi},\ Q''_i = g^{t_i},\quad T_i = E_i^{\hat{s}_i}.$$

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

$$\boldsymbol{R}_i = G_i^{s_i\boldsymbol{v}_i},\ \ \boldsymbol{R}'_i = G_i^{\kappa s_i\boldsymbol{v}_i},$$
$$Q_i = g^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)},\ \ Q'_i = (f\prod_{j'\in\bar{R}_i} f_{j'})^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)} Z_i^{t_i} f^{\pi},\ \ Q''_i = g^{t_i},\quad T_i = M\cdot E_i^{\tau s_i(\boldsymbol{v}_i\cdot\boldsymbol{v}_c)}.$$

2. For each column $j\in[m]$:
   - if $j<\bar{j}$: randomly chooses $\mu_j\in\mathbb{Z}_p$, and sets $\boldsymbol{C}_j = H_j^{\tau(\boldsymbol{v}_c+\mu_j\boldsymbol{\chi}_3)}\cdot g^{\kappa\boldsymbol{w}_j},\ \ \boldsymbol{C}'_j = g^{\boldsymbol{w}_j}$.
   - if $j\geq\bar{j}$: sets $\boldsymbol{C}_j = H_j^{\tau\boldsymbol{v}_c}\cdot g^{\kappa\boldsymbol{w}_j},\ \ \boldsymbol{C}'_j = g^{\boldsymbol{w}_j}$.
3. $P = g^{\pi},\ \{P_x = g^{\delta_x},\ \ P'_x = (H^x h)^{\delta_x} G^{-\pi}\}_{\forall x\in S}$.

$\mathsf{Decrypt}_{\mathsf{A}}(\mathsf{PP}, CT_{R,S}, \mathsf{SK}_{(i,j),(A,\rho)}) \to M$ or $\perp$. For ciphertext $CT_{R,S} = \langle R, S, (\boldsymbol{R}_i, \boldsymbol{R}'_i, Q_i, Q'_i, Q''_i, T_i)_{i=1}^m, (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^m,\ P, \{P_x, P'_x\}_{x\in S}\rangle$ and private key $\mathsf{SK}_{(i,j),(A,\rho)} = \Big((i,j),(A,\rho),\ K_{i,j}, K'_{i,j}, K''_{i,j}, \{\bar{K}_{i,j,j'}\}_{j'\in[m]\setminus\{j\}},\ \{K_{i,j,k,1},\ K_{i,j,k,2}, K_{i,j,k,3}\}_{k\in[l]}\Big)$, if $(i,j)\in R$ or S does not satisfy (A,ρ), the algorithm outputs $\perp$, otherwise:

1. Since S satisfies (A, ρ), the algorithm can efficiently compute constants $\{\omega_k \in \mathbb{Z}_p\}$ such that $\sum_{\rho(k)\in S} \omega_k A_k = (1, 0, \ldots, 0)$, then compute

$$\begin{aligned} D_P &= \prod_{\rho(k)\in S} \big(e(K_{i,j,k,1}, P) \cdot e(K_{i,j,k,2}, P_{\rho(k)}) \cdot e(K_{i,j,k,3}, P'_{\rho(k)})\big)^{\omega_k} \\ &= \prod_{\rho(k)\in S} \big(e(f^{(A_k\cdot \boldsymbol{u})} G^{\xi_k}, g^{\pi}) \cdot e((H^{\rho(k)}h)^{-\xi_k}, g^{\delta_{\rho(k)}}) \cdot e(g^{\xi_k}, (H^{\rho(k)}h)^{\delta_{\rho(k)}} G^{-\pi})\big)^{\omega_k} \\ &= \prod_{\rho(k)\in S} \big(e(f^{(A_k\cdot \boldsymbol{u})}, g^{\pi})\big)^{\omega_k} = e(f, g)^{\pi\sigma_{i,j}}. \end{aligned}$$

Note that if S does not satisfy (A, ρ), no such constants $\{\omega_k \in \mathbb{Z}_p\}$ would exist.
2. Since $(i, j) \in \bar{R}(= [m, m] \setminus R)$ implies $j \in \bar{R}_i$, the algorithm can compute

$$\bar{K}_{i,j} = K_{i,j} \cdot \Big(\prod_{j'\in\bar{R}_i\setminus\{j\}} \bar{K}_{i,j,j'}\Big) = g^{\alpha_i} g^{r_i c_j} (f f_j)^{\sigma_{i,j}} \cdot \Big(\prod_{j'\in\bar{R}_i\setminus\{j\}} f_{j'}^{\sigma_{i,j}}\Big) = g^{\alpha_i} g^{r_i c_j} \cdot \Big(f \prod_{j'\in\bar{R}_i} f_{j'}\Big)^{\sigma_{i,j}}.$$

Note that if $(i, j) \in R$ (implying $j \notin \bar{R}_i$), the algorithm cannot produce such a $\bar{K}_{i,j}$. The algorithm then computes

$$D_I = \frac{e(\bar{K}_{i,j}, Q_i) \cdot e(K''_{i,j}, Q''_i)}{e(K'_{i,j}, Q'_i)} \cdot \frac{e_3(\boldsymbol{R}'_i, \boldsymbol{C}'_j)}{e_3(\boldsymbol{R}_i, \boldsymbol{C}_j)}.$$

3. Computes $M = T_i/(D_P \cdot D_I)$. Suppose that the ciphertext is generated from message M' and encryption index $(\bar{i}, \bar{j})$, it can be verified that only when $(i > \bar{i})$ or $(i = \bar{i} \wedge j \geq \bar{j})$, $M = M'$. This is because 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 is also referred to Appendix B.

G.3 Augmented R-KP-ABE Security

The following theorem states that the AugR-KP-ABE proposed above is message-hiding. Then in Theorem 9, we state that the AugR-KP-ABE is also selectively index-hiding.

Theorem 8. *No PPT adversary can win* $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}}$ *with non-negligible advantage.*

Proof. The argument for message-hiding in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{MH}}$ is straightforward since an encryption to index $N+1$ (i.e. $(m+1, 1)$) contains no information about the message. The simulator simply runs $\mathsf{Setup}_{\mathsf{A}}$ and $\mathsf{KeyGen}_{\mathsf{A}}$ and encrypts M_b under the challenge (revocation list, attribute set) pair (R^*, S^*) and index $(m + 1, 1)$. Since for all $i = 1$ to m, $T_i = E_i^{\hat{s}_i}$ contains no information about the message, the bit b is perfectly hidden and $\mathsf{MH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}} = 0$.

Theorem 9. *Suppose that the D3DH, the DLIN and the Source Group q-Parallel DBDH Assumption hold. Then no PPT adversary can selectively win* $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$ *with non-negligible advantage, provided that the size of the challenge attribute set is* $\leq q$.

Proof. It follows Lemma 4 and Lemma 5 below.

Lemma 4. *If the D3DH and the Source Group q-Parallel DBDH Assumption hold, then for $\bar{j} < m$, no PPT adversary can selectively distinguish between an encryption to $(\bar{i}, \bar{j})$ and $(\bar{i}, \bar{j}+1)$ in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$ with non-negligible advantage, provided that the size of the challenge attribute set is $\leq q$.*

Proof. In $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$ with index $(\bar{i}, \bar{j})$, let (R^*, S^*) be the challenge (revocation list, attribute set) pair, the restriction is that the adversary $\mathcal{A}$ does not query a decryption key for (index, access policy) pair $((i,j), \mathbb{A}_{(i,j)})$ such that $\big((i,j) = (\bar{i}, \bar{j})\big) \wedge \big((i,j) \in [m,m] \setminus R^*\big) \wedge \big(S^* \; satisfies \; \mathbb{A}_{(i,j)}\big)$. Under this restriction, there are two ways for $\mathcal{A}$ to take:

Case I: In Phase 1 and Phase 2, $\mathcal{A}$ does not query a decryption key with index $(\bar{i}, \bar{j})$.
Case II: In Phase 1 or Phase 2, $\mathcal{A}$ queries a decryption key with index $(\bar{i}, \bar{j})$. Let $\mathbb{A}_{(\bar{i},\bar{j})}$ be the corresponding access policy. **Case II** has the following sub-cases:

1. $(\bar{i}, \bar{j}) \notin [m,m] \setminus R^*$, S^* satisfies $\mathbb{A}_{(\bar{i},\bar{j})}$.
2. $(\bar{i}, \bar{j}) \notin [m,m] \setminus R^*$, S^* does not satisfy $\mathbb{A}_{(\bar{i},\bar{j})}$.
3. $(\bar{i}, \bar{j}) \in [m,m] \setminus R^*$, S^* does not satisfy $\mathbb{A}_{(\bar{i},\bar{j})}$.

If $\mathcal{A}$ is in **Case I**, **Case II.1** or **Case II.2**, it follows the restrictions in the index-hiding game for Augmented Broadcast Encryption (AugBE) in [9], where the adversary does not query the key with index $(\bar{i}, \bar{j})$ or $(\bar{i}, \bar{j})$ is not in the receiver list $[m,m] \setminus R^*$. **Case II.3** captures the index-hiding requirement of Augmented R-KP-ABE in that even if a user has a key with index $(\bar{i}, \bar{j})$ and $(\bar{i}, \bar{j}) \notin R^*$, the user cannot distinguish between an encryption to $(R^*, S^*, (\bar{i}, \bar{j}))$ and $(R^*, S^*, (\bar{i}, \bar{j}+1))$ if the corresponding access policy $\mathbb{A}_{(\bar{i},\bar{j})}$ is not satisfied by S^*. This is the most challenging part of proving the index-hiding when we attempt to *securely intertwine* the tracing techniques of broadcast encryption (e.g. [9]) into the large universe KP-ABE (e.g. [26]). Compared to the proof of [20], the challenge here is to prove the index-hiding in the large universe setting, as discussed previously.

To prove this lemma, we flip a random coin $c \in \{0,1\}$ as our guess on which case that $\mathcal{A}$ is in. If $\mathcal{A}$ is in **Case I**, **Case II.1** or **Case II.2**, we make a reduction that uses $\mathcal{A}$ to solve a D3DH problem instance, using a proof technique similar to that of [9]. Actually, in this proof, we reduce from our AugR-KP-ABE to the AugBE in [9]. If $\mathcal{A}$ is in **Case I**, **Case II.2** or **Case II.3**, we use $\mathcal{A}$ to solve a Source Group q-Parallel DBDH problem instance, which is where the main novelty resides among all the proofs in this work. Please refer to Appendix I for details.

Lemma 5. *If the D3DH, the DLIN and the Source Group q-Parallel DBDH Assumption hold, then for $1 \leq \bar{i} \leq m$, no PPT adversary can selectively distinguish between an encryption to $(\bar{i}, m)$ and $(\bar{i}+1, 1)$ in $\mathsf{Game}^{\mathsf{A}}_{\mathsf{IH}}$ with non-negligible advantage, provided that the size of the challenge attribute set is $\leq q$.*

Proof. Similar to the proof of Lemma 6.3 in [9], to prove this lemma we define the following hybrid experiment: H_1: encrypt to $(\bar{i}, \bar{j} = m)$; H_2: encrypt to $(\bar{i}, \bar{j} = m+1)$; and H_3: encrypt to $(\bar{i}+1, 1)$. This lemma follows Claim 3 and Claim 4 below.

Claim 3. *If the D3DH and the Source Group q-Parallel DBDH Assumption hold, then no PPT adversary can selectively distinguish between experiment H_1 and H_2 with non-negligible advantage, provided that the size of the challenge attribute set is $\leq q$.*

Proof. The proof is identical to that for Lemma 4.

Claim 4. *If the D3DH and the DLIN hold, then no PPT adversary can distinguish between experiment H_2 and H_3 with non-negligible advantage.*

Proof. Note that $(\bar{i}, m+1) \notin [m,m]$ implies that for any revocation list $R^* \subseteq [m,m]$, we have $(\bar{i}, m+1) \notin \bar{R}^* (= [m,m] \setminus R^*)$, i.e, the adversaries for distinguishing H_2 and H_3 will not be in **Case II.3**. Thus, we can prove this claim in a similar way to that of [9]. Actually, in this proof, we reduce from our AugR-KP-ABE to the AugBE in [9]. In the proof of index-hiding for an AugBE scheme Σ_{AugBE} in [9, Lemma 6.3], two hybrid experiments were defined and proven indistinguishable via a sequence of hybrid sub-experiments.

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

By following [9, Lemma 6.3], *if the D3DH and the DLIN hold, no PPT adversary can distinguish between* H_2^{AugBE} *and* H_3^{AugBE} *with non-negligible advantage for* Σ_{AugBE}. Suppose there is a PPT adversary $\mathcal{A}$ that can distinguish between H_2 and H_3 for Σ_{A} with non-negligible advantage. We can build a reduction, which is similar to that of **Case A** in Appendix I, to use $\mathcal{A}$ to distinguish between H_2^{AugBE} and H_3^{AugBE} for Σ_{AugBE} with non-negligible advantage.

H AugR-KP-ABE Implies Secure R-KP-ABE

To prove that the R-KP-ABE scheme Σ in Sec. F.2 is secure it remains to prove that Equation (4) holds for all $\bar{k} = 1, \ldots, N$. Consider a specific $\bar{k} \in [N]$. Adversary $\mathcal{B}$ plays the index-hiding game $\mathsf{Game}_{\mathsf{IH}}^{\mathsf{A}}$ with input $\bar{k}$ and works as follows:

Setup. $\mathcal{B}$ receives PP from its challenger in the index-hiding game $\mathsf{Game}_{\mathsf{IH}}^{\mathsf{A}}$. $\mathcal{B}$ runs adversary $\mathcal{A}$ in the extend message-hiding game $\mathsf{Game}_{\mathsf{EMH}}^{\mathsf{A}}$ and gives PP to $\mathcal{A}$.

Phase 1. For $i = 1$ to Q_1, $\mathcal{A}$ adaptively submits (index, access policy) pair $(k_i, \mathbb{A}_{k_i})$ to $\mathcal{B}$. $\mathcal{B}$ submits $(k_i, \mathbb{A}_{k_i})$ to the challenger and receives secret key $\mathsf{SK}_{k_i,\mathbb{A}_{k_i}}$. Then $\mathcal{B}$ gives $\mathsf{SK}_{k_i,\mathbb{A}_{k_i}}$ to $\mathcal{A}$.

Challenge. $\mathcal{A}$ submits two equal-length messages M_0, M_1 and a (revocation list, attribute set) (R^*, S^*) to $\mathcal{B}$, under the restriction that none of the queried pairs $\{(k_i, \mathbb{A}_{k_i})\}_{i=1}^{Q_1}$ can satisfy $(k_i \in [N] \setminus R^*) \wedge (S^* \; satisfies \; \mathbb{A}_{k_i})$. $\mathcal{B}$ flips a coin $\gamma \in \{0,1\}$, then gives M_γ and (R^*, S^*) to its challenger. Note that (R^*, S^*) satisfies the restriction on $\mathcal{B}$ in $\mathsf{Game}_{\mathsf{IH}}^{\mathsf{A}}$ that none of the queried pairs $\{(k_i, \mathbb{A}_{k_i})\}_{i=1}^{Q_1}$ can satisfy $(k_i = \bar{k}) \wedge (k_i \in [N] \setminus R^*) \wedge (S^* \; satisfies \; \mathbb{A}_{k_i})$. $\mathcal{B}$ receives $CT_{R^*,S^*} \leftarrow \mathsf{Encrypt}_{\mathsf{A}}(\mathsf{PP}, M_\gamma, R^*, S^*, \bar{k}+b)$ for some random $b \in \{0,1\}$. Then $\mathcal{B}$ gives CT_{R^*,S^*} to $\mathcal{A}$.

Phase 2. For $i = Q_1 + 1$ to Q, $\mathcal{A}$ adaptively submits (index, access policy) pair $(k_i, \mathbb{A}_{k_i})$ to $\mathcal{B}$, under the restriction that $(k_i, \mathbb{A}_{k_i})$ does not satisfy $(k_i \in [N] \setminus R^*) \wedge (S^* \; satisfies \; \mathbb{A}_{k_i})$. $\mathcal{B}$ submits $(k_i, \mathbb{A}_{k_i})$ to the challenger. Note that $(k_i, \mathbb{A}_{k_i})$ satisfies the the restriction on $\mathcal{B}$ in $\mathsf{Game}_{\mathsf{IH}}^{\mathsf{A}}$ that $(k_i, \mathbb{A}_{k_i})$ does not satisfy $(k_i = \bar{k}) \wedge (k_i \in [N] \setminus R^*) \wedge (S^* \; satisfies \; \mathbb{A}_{k_i})$. $\mathcal{B}$ receives secret key $\mathsf{SK}_{k_i,\mathbb{A}_{k_i}}$ from the challenger. Then $\mathcal{B}$ gives $\mathsf{SK}_{k_i,\mathbb{A}_{k_i}}$ to $\mathcal{A}$.

Guess. $\mathcal{A}$ outputs a guess $\gamma' \in \{0,1\}$ for γ. If $\gamma' = \gamma$ then $\mathcal{B}$ returns 0 to its challenger. Otherwise $\mathcal{B}$ returns 1 to its challenger.

Now, observe that when $b = 0$ then $\mathcal{B}$ is emulating perfectly an $\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}]$ challenger. When $b = 1$ then $\mathcal{B}$ is emulating perfectly an $\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}+1]$ challenger. A standard argument now shows that $|\mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}] - \mathsf{EMH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{A}}[\bar{k}+1]| \leq 2 \cdot \mathsf{IH}^{\mathsf{A}}\mathsf{Adv}_{\mathcal{B}}[\bar{k}]$ as required.

I Proof of Lemma 4

Proof. Suppose there exists a PPT adversary $\mathcal{A}$ that selectively breaks the index-hiding game with non-negligible advantage $Adv_{\mathcal{A}}$. We construct a PPT algorithm $\mathcal{B}$, which is given a D3DH problem

instance and a Source Group q-parallel DBDH problem instance, and solves at least one of the two problems with non-negligible advantage. $\mathcal{B}$ flips a random coin $c \in \{0,1\}$, if $c=0$, $\mathcal{B}$ interacts with $\mathcal{A}$ in **Case A** as guessing "$\mathcal{A}$ is not in **Case II.3**", otherwise $\mathcal{B}$ interacts with $\mathcal{A}$ in **Case B** as guessing "$\mathcal{A}$ is not in **Case II.1**".

Case A: **$\mathcal{B}$ uses $\mathcal{A}$ to solve the D3DH problem.** Garg et al. [9, Sec. 5.1] proposed an AugBE scheme $\Sigma_{\mathsf{AugBE}} = (\mathsf{Setup}_{\mathsf{AugBE}}, \mathsf{Encrypt}_{\mathsf{AugBE}}, \mathsf{Decrypt}_{\mathsf{AugBE}})$ and proved that it is index-hiding. The Lemma 6.2 of [9] states that *if the D3DH assumption holds, then for $\bar{j} < m$ no PPT adversary can distinguish between an encryption to $(\bar{i}, \bar{j})$ and $(\bar{i}, \bar{j}+1)$ in the index-hiding game for Σ_{AugBE} with non-negligible probability.* Note that if $\mathcal{A}$ is in **Case I**, **Case II.1** or **Case II.2**, it also follows the restrictions of the index-hiding game for Σ_{AugBE}, here we do not build a direct reduction that uses $\mathcal{A}$ to solve the D3DH problem, instead, we build a reduction to break the index-hiding property of Σ_{AugBE}. We first give the reduction sketch below.

First we review the structures of public key $\mathsf{PK}^{\mathsf{AugBE}}$, private key $\mathsf{SK}^{\mathsf{AugBE}}_{(i,j)}$ and ciphertext $CT^{\mathsf{AugBE}}_{\bar{R}}$ of Σ_{AugBE} [9][6].

$$\mathsf{PK}^{\mathsf{AugBE}} = \big(\ g,\ \{E_i = e(g,g)^{\alpha_i},\ G_i = g^{r_i}\}_{i\in[m]},\ \{H_j = g^{c_j},\ f_j\}_{j\in[m]}\ \big),$$
$$\mathsf{SK}^{\mathsf{AugBE}}_{(i,j)} = \big(K_{i,j},\ K'_{i,j},\ \ \{\bar{K}_{i,j,j'}\}_{j'\in[m]\setminus\{j\}}\ \big) = \big(g^{\alpha_i} g^{r_i c_j} f_j^{\sigma_{i,j}},\ g^{\sigma_{i,j}},\ \ \{f_{j'}^{\sigma_{i,j}}\}_{j'\in[m]\setminus\{j\}}\ \big),$$
$$CT^{\mathsf{AugBE}}_{\bar{R}} = \langle (\boldsymbol{R}_i, \boldsymbol{R}'_i, Q_i, Q'_i, T_i)_{i=1}^{m},\ (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^{m},\ \bar{R}\rangle,$$

where $CT^{\mathsf{AugBE}}_{\bar{R}}$ is for receiver list $\bar{R}$ and index (i^*, j^*) with

1. For each $i \in [m]$:
   - if $i < i^*$: $\boldsymbol{R}_i = g^{\boldsymbol{v}_i},\ \ \boldsymbol{R}'_i = g^{\kappa \boldsymbol{v}_i},\ \ Q_i = g^{s_i},\ \ Q'_i = (\prod_{j'\in\bar{R}_i} f_{j'})^{s_i},\ \ T_i = E_i^{\hat{s}_i}$.
   - if $i \geq i^*$: $\boldsymbol{R}_i = G_i^{s_i \boldsymbol{v}_i},\ \boldsymbol{R}'_i = G_i^{\kappa s_i \boldsymbol{v}_i},\ Q_i = g^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)},\ \ Q'_i = (\prod_{j'\in\bar{R}_i} f_{j'})^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)},\ T_i = M \cdot E_i^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}$.
2. For each $j \in [m]$:
   - if $j < j^*$: $\boldsymbol{C}_j = H_j^{\tau(\boldsymbol{v}_c + \mu_j \boldsymbol{\chi}_3)} \cdot g^{\kappa \boldsymbol{w}_j},\ \ \boldsymbol{C}'_j = g^{\boldsymbol{w}_j}$.
   - if $j \geq j^*$: $\boldsymbol{C}_j = H_j^{\tau \boldsymbol{v}_c} \cdot g^{\kappa \boldsymbol{w}_j},\ \ \boldsymbol{C}'_j = g^{\boldsymbol{w}_j}$.

Setup. From the received $\mathsf{PK}^{\mathsf{AugBE}}$, $\mathcal{B}$ generates PP for $\mathcal{A}$ by randomly choosing β, θ, z_i $(i \in [m]) \in \mathbb{Z}_p$ and $h, H \in \mathbb{G}$, and setting $f = g^{\beta}, G = g^{\theta}, \{Z_i = g^{z_i}\}_{i\in[m]}$.
Phase 1 and **2**. As $\mathcal{B}$ can compute $f^{\sigma_{i,j}} = (g^{\sigma_{i,j}})^{\beta}$ and $Z_i^{\sigma_{i,j}} = (g^{\sigma_{i,j}})^{z_i}$ without $\sigma_{i,j}$, $\mathcal{B}$ can produce $\mathsf{SK}_{(i,j),(A,\rho)}$ for $\mathcal{A}$, using $\mathsf{SK}^{\mathsf{AugBE}}_{(i,j)}$ and random $u_2, \ldots, u_n \in \mathbb{Z}_p$, $\{\xi_k \in \mathbb{Z}_p\}_{k\in[l]}$.
Challenge. As $\mathcal{B}$ can compute $f^{s_i} = (g^{s_i})^{\beta}$ and $f^{\tau s_i (\boldsymbol{v_i} \cdot \boldsymbol{v}_c)} = (g^{\tau s_i (\boldsymbol{v_i} \cdot \boldsymbol{v}_c)})^{\beta}$ without s_i or $\tau s_i (\boldsymbol{v_i} \cdot \boldsymbol{v}_c)$, by using its challenge ciphertext $CT^{\mathsf{AugBE}}_{\bar{R}^*}$ (for $\bar{R}^* = [m,m] \setminus R^*$) and random $\pi, t_1, \ldots, t_m, \delta_x (x \in S^*) \in \mathbb{Z}_p$, $\mathcal{B}$ can produce the challenge ciphertext CT_{R^*,S^*} for $\mathcal{A}$.
Guess. $\mathcal{B}$ sends $\mathcal{A}$'s guess $b' \in \{0,1\}$ to its challenger.

During the interaction, if $\mathcal{A}$ is in **Case II.3**, $\mathcal{B}$ will abort and return a random $b \in \{0,1\}$ to its challenger.
Now we give the reduction details.

Init. The adversary $\mathcal{A}$ gives $\mathcal{B}$ the challenge attribute set S^*.

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

Setup. The challenger gives $\mathcal{B}$ the public key $\mathsf{PK}^{\mathsf{AugBE}}$

$$\mathsf{PK}^{\mathsf{AugBE}} = (\ g,\ \{E_i = e(g,g)^{\alpha_i},\ G_i = g^{r_i}\}_{i\in[m]},\ \{H_j = g^{c_j},\ f_j\}_{j\in[m]}\ ),$$

and private keys $\{\mathsf{SK}^{\mathsf{AugBE}}_{(i,j)}\}_{(i,j)\in[m,m]\setminus\{(\bar{i},\bar{j})\}}$ as

$$\mathsf{SK}^{\mathsf{AugBE}}_{(i,j)} = \big(\tilde{K}_{i,j},\ \tilde{K}'_{i,j},\ \ \{\tilde{\tilde{K}}_{i,j,j'}\}_{j'\in[m]\setminus\{j\}}\ \big) = \big(g^{\alpha_i} g^{r_i c_j} f_j^{\sigma_{i,j}},\ g^{\sigma_{i,j}},\ \ \{f_{j'}^{\sigma_{i,j}}\}_{j'\in[m]\setminus\{j\}}\ \big),$$

where $g, f_1, \ldots, f_m \in \mathbb{G}$ and $\{\alpha_i, r_i \in \mathbb{Z}_p\}_{i\in[m]}$, $\{c_j \in \mathbb{Z}_p\}_{j\in[m]}$, $\{\sigma_{i,j} \in \mathbb{Z}_p\}_{(i,j)\in[m,m]\setminus\{(\bar{i},\bar{j})\}}$ are randomly chosen. $\mathcal{B}$ sets $\tilde{c} = 0$ to denote that $\mathcal{B}$ does not obtain the private $\mathsf{SK}^{\mathsf{AugBE}}_{(\bar{i},\bar{j})}$.

$\mathcal{B}$ randomly chooses $\beta, \theta, z_1, \ldots, z_m \in \mathbb{Z}_p$ and $h, H \in \mathbb{G}$, then gives $\mathcal{A}$ the following public parameter PP:

$$\big(g,\ h,\ f = g^{\beta},\ f_1, \ldots, f_m,\ G = g^{\theta},\ H,\ \ \{E_i,\ G_i,\ Z_i = g^{z_i}\}_{i\in[m]},\ \ \{H_j\}_{j\in[m]}\big).$$

Phase 1. $\mathcal{A}$ adaptively submits $((i,j),(A,\rho))$ to $\mathcal{B}$, where (A,ρ) is an LSSS matrix. Let A be an $l \times n$ matrix. If $(i,j) = (\bar{i},\bar{j})$, then $\mathcal{B}$ sets $\tilde{c} = 1$ and submits $\tilde{c}$ to its challenger, and receives the private key $\mathsf{SK}^{\mathsf{AugBE}}_{(\bar{i},\bar{j})}$. $\mathcal{B}$ randomly chooses $u_2, \ldots, u_n \in \mathbb{Z}_p$ and $\{\xi_k \in \mathbb{Z}_p\}_{k\in[l]}$, and sets the value of $\boldsymbol{u} \in \mathbb{Z}_p^n$ by implicitly setting $\boldsymbol{u} = (\sigma_{i,j}, u_2, \ldots, u_n)$. For $k = 1$ to l, let $A_k = (A_{k,1}, \ldots, A_{k,n}) \in \mathbb{Z}_p^n$ be the k^{th} row of A. $\mathcal{B}$ creates a private key $\mathsf{SK}_{(i,j),(A,\rho)} = \big((i,j),(A,\rho),\ K_{i,j}, K'_{i,j}, K''_{i,j},\ \{\bar{K}_{i,j,j'}\}_{j'\in[m]\setminus\{j\}},\ \{K_{i,j,k,1},\ K_{i,j,k,2}, K_{i,j,k,3}\}_{k\in[l]}\big)$ from $\mathsf{SK}^{\mathsf{AugBE}}_{(i,j)}$ as follows:

$$K_{i,j} = \tilde{K}_{i,j} \cdot (\tilde{K}'_{i,j})^{\beta},\ \ K'_{i,j} = \tilde{K}'_{i,j},\ \ K''_{i,j} = (\tilde{K}'_{i,j})^{z_i},\ \ \{\bar{K}_{i,j,j'} = \tilde{\tilde{K}}_{i,j,j'}\}_{j'\in[m]\setminus\{j\}},$$
$$\{K_{i,j,k,1} = (\tilde{K}'_{i,j})^{\beta A_{k,1}} f^{\sum_{d=2}^{n} u_d A_{k,d}} G^{\xi_k},\ K_{i,j,k,2} = (H^{\rho(k)} h)^{-\xi_k},\ K_{i,j,k,3} = g^{\xi_k}\}_{k\in[l]}.$$

Challenge. $\mathcal{A}$ submits a message M and a revocation list R^*. $\mathcal{B}$ sets $\bar{R}^* = [m,m] \setminus R^*$.

- if $(\bar{i},\bar{j}) \in \bar{R}^* \wedge \tilde{c} = 1$: $\mathcal{A}$ is in **Case II.3**. $\mathcal{B}$ returns a random $\beta_3 \in \{0,1\}$ to its challenger, then aborts.
- if $(\bar{i},\bar{j}) \in \bar{R}^* \wedge \tilde{c} = 0$: $\mathcal{B}$ continues the following interaction.
- if $(\bar{i},\bar{j}) \notin \bar{R}^* \wedge \tilde{c} = 1$: $\mathcal{B}$ continues the following interaction.
- if $(\bar{i},\bar{j}) \notin \bar{R}^* \wedge \tilde{c} = 0$: $\mathcal{B}$ sets $\tilde{c} = 1$ and submits c_1 to its challenger, and receives the private key $\mathsf{SK}^{\mathsf{AugBE}}_{(\bar{i},\bar{j})}$. Then $\mathcal{B}$ continues the following interaction.

Now $\mathcal{B}$ ends the Query Phase for the AugBE index-hiding game with its challenger, and submits $(M, \bar{R}^*)$ to the challenger. Note that from the view of the challenger, $\mathcal{B}$'s behaviors satisfy the restrictions in the AugBE index-hiding game, i.e., if $\mathcal{B}$ sends $\tilde{c} = 1$ to the challenger and obtains $\mathsf{SK}^{\mathsf{AugBE}}_{(\bar{i},\bar{j})}$ then $(\bar{i},\bar{j}) \notin \bar{R}^*$. The challenger gives $\mathcal{B}$ the challenge ciphertext $CT^{\mathsf{AugBE}}_{\bar{R}^*} = \langle(\tilde{\boldsymbol{R}}_i, \tilde{\boldsymbol{R}}'_i, \tilde{Q}_i, \tilde{Q}'_i, \tilde{T}_i)_{i=1}^m, (\tilde{\boldsymbol{C}}_j, \tilde{\boldsymbol{C}}'_j)_{j=1}^m,\ \bar{R}^*\rangle$, which is encrypted to $(i^*, j^*) \in \{(\bar{i},\bar{j}), (\bar{i},\bar{j}+1)\}$ and in the form of

1. For each $i \in [m]$:
   - if $i < i^*$: $\tilde{\boldsymbol{R}}_i = g^{\boldsymbol{v}_i},\ \ \tilde{\boldsymbol{R}}'_i = g^{\kappa \boldsymbol{v}_i},\ \ \tilde{Q}_i = g^{s_i},\ \ \tilde{Q}'_i = (\prod_{j'\in\bar{R}_i^*} f_{j'})^{s_i},\ \ \tilde{T}_i = E_i^{\hat{s}_i}$.
   - if $i \geq i^*$: $\tilde{\boldsymbol{R}}_i = G_i^{s_i \boldsymbol{v}_i},\ \tilde{\boldsymbol{R}}'_i = G_i^{\kappa s_i \boldsymbol{v}_i},\ \tilde{Q}_i = g^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)},\ \ \tilde{Q}'_i = (\prod_{j'\in\bar{R}_i^*} f_{j'})^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)},\ \tilde{T}_i = M \cdot E_i^{\tau s_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}$.
2. For each $j \in [m]$:

– if $j < j^*$: $\tilde{\boldsymbol{C}}_j = H_j^{\tau(\boldsymbol{v}_c + \mu_j \boldsymbol{\chi}_3)} \cdot g^{\kappa \boldsymbol{w}_j}$, $\tilde{\boldsymbol{C}}'_j = g^{\boldsymbol{w}_j}$.
– if $j \geq j^*$: $\tilde{\boldsymbol{C}}_j = H_j^{\tau \boldsymbol{v}_c} \cdot g^{\kappa \boldsymbol{w}_j}$, $\tilde{\boldsymbol{C}}'_j = g^{\boldsymbol{w}_j}$.

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

$\mathcal{B}$ randomly chooses $\pi, t_1, \ldots, t_m, \delta_x (x \in S^*) \in \mathbb{Z}_p$, then creates the ciphertext $\langle R^*, S^*, (\boldsymbol{R}_i, \boldsymbol{R}'_i, Q_i, Q'_i, Q''_i, T_i)_{i=1}^m, (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^m, P, \{P_x, P'_x\}_{x \in S^*} \rangle$ as follows:

1. For each $i \in [m]$: $\boldsymbol{R}_i = \tilde{\boldsymbol{R}}_i$, $\boldsymbol{R}'_i = \tilde{\boldsymbol{R}}'_i$, $Q_i = \tilde{Q}_i$, $Q'_i = \tilde{Q}_i^{\beta} \cdot \tilde{Q}'_i \cdot Z_i^{t_i} f^{\pi}$, $Q''_i = g^{t_i}$, $T_i = \tilde{T}_i$.
2. For each $j \in [m]$: $\boldsymbol{C}_j = \tilde{\boldsymbol{C}}_j$, $\boldsymbol{C}'_j = \tilde{\boldsymbol{C}}'_j$.
3. $P = g^{\pi}$, $\{P_x = g^{\delta_x}, P'_x = (H^x h)^{\delta_x} G^{-\pi}\}_{x \in S^*}$.

Phase 2. $\mathcal{A}$ adaptively submits $((i, j), (A, \rho))$ to $\mathcal{B}$.

– if $(i, j) \neq (\bar{i}, \bar{j})$: $\mathcal{B}$ creates the private key $\mathsf{SK}_{(i,j),(A,\rho)}$ from $\mathsf{SK}_{(i,j)}^{\mathsf{AugBE}}$ as in **Phase 1**.
– if $(i, j) = (\bar{i}, \bar{j}) \wedge \tilde{c} = 1$: this implies $\mathcal{B}$ has obtained $\mathsf{SK}_{(\bar{i},\bar{j})}^{\mathsf{AugBE}}$ from its challenger. $\mathcal{B}$ creates the private key $\mathsf{SK}_{(\bar{i},\bar{j}),(A,\rho)}$ from $\mathsf{SK}_{(\bar{i},\bar{j})}^{\mathsf{AugBE}}$ as in **Phase 1**.
– if $(i, j) = (\bar{i}, \bar{j}) \wedge \tilde{c} = 0$: observing $\mathcal{B}$'s behaviors in **Challenge** phase, we have that $\tilde{c} = 0$ implies $(\bar{i}, \bar{j}) \in \bar{R}^*$. In other words, $\mathcal{A}$ is querying a key with index $(\bar{i}, \bar{j})$ and $(\bar{i}, \bar{j}) \in \bar{R}^*$, i.e., $\mathcal{A}$ is in **Case II.3**. $\mathcal{B}$ return a random $\beta_3 \in \{0, 1\}$ to its challenger, then aborts.

Guess. $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ to $\mathcal{B}$, then $\mathcal{B}$ sets $\beta_3 = b'$ ands returns β_3 to its challenger.

When $\mathcal{B}$ does not abort, $\mathcal{B}$'s advantage in the index-hiding game for Σ_{AugBE} will be exactly equal to $\mathcal{A}$'s advantage in the index-hiding game for our AugR-KP-ABE scheme Σ_{A}. Thus, $\mathcal{B}$'s final advantage in the index-hiding game for Σ_{AugBE} is $Adv_{\mathcal{B},3} = Adv_{\mathcal{A}} \cdot \Pr[\mathcal{A} \text{ is not in } \textbf{Case II.3} \wedge (c = 0)]$.

Case B: **$\mathcal{B}$ uses $\mathcal{A}$ to solve the Source Group q-parallel DBDH problem.** $\mathcal{B}$ is given

$$D = \big((p, \mathbb{G}, \mathbb{G}_T, e), g, g^a, g^b, g^{cd}, g^d, g^{ad}, g^{(acd)^2}, \\ g^{d_i},\ g^{acdd_i},\ g^{a^2 cdd_i},\ g^{acd/d_i},\ g^{b/d_i^2},\ g^{b^2/d_i^2} \quad \forall i \in [q], \\ g^{acdd_i/d_j},\ g^{(acd)^2 d_i/d_j},\ g^{bd_i/d_j^2},\ g^{abcdd_i/d_j^2} \quad \forall i, j \in [q]\ s.t.\ i \neq j\ \big)$$

and T, where $(p, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}$, $g \xleftarrow{R} \mathbb{G}$, $a, b, c, d, d_1, \ldots, d_q \xleftarrow{R} \mathbb{Z}_p$, and T is either equal to g^{abc} or is a random element of $\mathbb{G}$. $\mathcal{B}$'s goal is to determine $T = g^{abc}$ or T is a random element from $\mathbb{G}$.

Init. $\mathcal{A}$ gives $\mathcal{B}$ the challenge attribute set $S^* = \{a_1^*, \ldots, a_{l^*}^*\} \subseteq \mathcal{U} = \mathbb{Z}_p$, where $|S^*| = l^* \leq q$.

Setup. $\mathcal{B}$ randomly chooses $\{\alpha_i \in \mathbb{Z}_p\}_{i \in [m]}$, $\{r_i,\ z'_i \in \mathbb{Z}_p\}_{i \in [m] \setminus \{\bar{i}\}}$, $r'_{\bar{i}}, z_{\bar{i}} \in \mathbb{Z}_p$, $\{c'_j \in \mathbb{Z}_p\}_{j \in [m]}$, and $\beta, \eta, \delta, \theta, \theta_1, \ldots, \theta_m \in \mathbb{Z}_p$. $\mathcal{B}$ gives $\mathcal{A}$ the public parameter PP:

$$\Big(g,\ h = g^{\beta} \cdot \big(\prod_{t \in [l^*]} g^{acd/d_t}\big) \cdot \big(\prod_{t \in [l^*]} (g^{b/d_t^2})^{-a_t^*}\big),\ f = (g^a)^{\theta},\ \{f_j = g^{\theta_j}\}_{j \in [m]}, \\ G = (g^a)^{\delta},\ H = g^{\eta} \cdot \big(\prod_{t \in [l^*]} g^{b/d_t^2}\big),\ \{E_i = e(g, g)^{\alpha_i}\}_{i \in [m]},$$

$$\{G_i = g^{r_i},\ Z_i = (g^a)^{z'_i}\}_{i\in[m]\setminus\{\bar{i}\}},\ \{H_j = (g^d)^{c'_j}\}_{j\in[m]\setminus\{\bar{j}\}},\quad G_{\bar{i}} = (g^a)^{r'_{\bar{i}}},\ Z_{\bar{i}} = g^{z_{\bar{i}}},\ H_{\bar{j}} = (g^b)^{c'_{\bar{j}}}.\ \Big).$$

Note that $\mathcal{B}$ implicitly chooses $r_{\bar{i}},\ z_i (i \in [m]\setminus\{\bar{i}\}),\ c_j (j \in [m])\ \in \mathbb{Z}_p$ such that

$$ar'_{\bar{i}} \equiv r_{\bar{i}} \bmod p,\ az'_i \equiv z_i \bmod p\ \forall i \in [m]\setminus\{\bar{i}\},$$
$$dc'_j \equiv c_j \bmod p\ \forall j \in [m]\setminus\{\bar{j}\},\quad bc'_{\bar{j}} \equiv c_{\bar{j}} \bmod p.$$

Phase 1. To respond to $\mathcal{A}$'s query for $((i,j),(A,\rho))$, let $l \times n$ be the size of A,

• if $(i,j) \neq (\bar{i},\bar{j})$: $\mathcal{B}$ randomly chooses $\boldsymbol{u} = (\sigma_{i,j}, u_2, \ldots, u_n) \in \mathbb{Z}_p^n$ and $\{\xi_k \in \mathbb{Z}_p\}_{k=1}^l$, and creates the decryption key $\mathsf{SK}_{(i,j),(A,\rho)}$:

$$K_{i,j} = \begin{cases} g^{\alpha_i}(g^d)^{r_i c'_j} f^{\sigma_{i,j}}, & : i \neq \bar{i}, j \neq \bar{j} \\ g^{\alpha_i}(g^{ad})^{r'_{\bar{i}} c'_j} f^{\sigma_{i,j}}, & : i = \bar{i}, j \neq \bar{j} \\ g^{\alpha_i}(g^b)^{r_i c'_{\bar{j}}} f^{\sigma_{i,j}}, & : i \neq \bar{i}, j = \bar{j}. \end{cases}$$
$$K'_{i,j} = g^{\sigma_{i,j}},\quad K''_{i,j} = Z_i^{\sigma_{i,j}},\quad \{\bar{K}_{i,j,j'} = f_{j'}^{\sigma_{i,j}}\}_{j'\in[m]\setminus\{j\}},$$
$$\{K_{i,j,k,1} = f^{(A_k\cdot\boldsymbol{u})} G^{\xi_k},\quad K_{i,j,k,2} = (H^{\rho(k)}h)^{-\xi_k},\ K_{i,j,k,3} = g^{\xi_k}\}_{k\in[l]}.$$

• if $(i,j) = (\bar{i},\bar{j})$: if S^* satisfies (A,ρ), then $\mathcal{A}$ is in **Case II.1**, $\mathcal{B}$ returns a random $\beta_q \in \{0,1\}$ to the challenger. Otherwise (i.e. S^* does not satisfy (A,ρ)), $\mathcal{B}$ first computes a vector $\bar{\boldsymbol{u}} = (\bar{u}_1, \ldots, \bar{u}_n) \in \mathbb{Z}_p^n$ that has first entry equal to 1 (i.e. $\bar{u}_1 = 1$) and is orthogonal to all of the rows A_k of A such that $\rho(k) \in S^*$ (i.e. $A_k \cdot \bar{\boldsymbol{u}} = 0\ \forall k \in [l]\ s.t.\ \rho(k) \in S^*$). Note that such a vector must exist since S^* fails to satisfy (A,ρ), and it is efficiently computable. Then $\mathcal{B}$ randomly chooses $\sigma'_{\bar{i},\bar{j}}, u'_2, \ldots, u'_n \in \mathbb{Z}_p$, $\{\xi_k \in \mathbb{Z}_p\}_{k\in[l]\ s.t.\ \rho(k)\in S^*}$, $\{\xi'_k \in \mathbb{Z}_p\}_{k\in[l]\ s.t.\ \rho(k)\notin S^*}$. Let $\boldsymbol{u}' = (0, u'_2, \ldots, u'_n) \in \mathbb{Z}_p^n$, $\mathcal{B}$ sets the values of $\sigma_{\bar{i},\bar{j}} \in \mathbb{Z}_p$, $\boldsymbol{u} \in \mathbb{Z}_p^n$, $\{\xi_k \in \mathbb{Z}_p\}_{k\in[l]\ s.t.\ \rho(k)\notin S^*}$ by implicitly setting

$$\sigma'_{\bar{i},\bar{j}} - br'_{\bar{i}}c'_{\bar{j}}/\theta \equiv \sigma_{\bar{i},\bar{j}} \bmod p,\quad \boldsymbol{u} = \boldsymbol{u}' + \sigma_{\bar{i},\bar{j}}\bar{\boldsymbol{u}},$$
$$\xi'_k + br'_{\bar{i}}c'_{\bar{j}}(A_k\cdot\bar{\boldsymbol{u}})/\delta - r'_{\bar{i}}c'_{\bar{j}} \sum_{t\in[l^*]} \frac{acdd_t(A_k\cdot\bar{\boldsymbol{u}})/\delta}{\rho(k) - a_t^*} \equiv \xi_k \bmod p\ \forall k \in [l]\ s.t.\ \rho(k) \notin S^*.$$

Note that for $a_t^* \in S^*$ and $\rho(k) \notin S^*$ we have $\rho(k) - a_t^* \neq 0$. $\mathcal{B}$ creates the private key $\mathsf{SK}_{(\bar{i},\bar{j}),(A,\rho)}$ as follows:

$$K_{\bar{i},\bar{j}} = g^{\alpha_{\bar{i}}} f^{\sigma'_{\bar{i},\bar{j}}} \big(g^{\sigma'_{\bar{i},\bar{j}}}(g^b)^{-r'_{\bar{i}}c'_{\bar{j}}/\theta}\big)^{\theta_j},\ K'_{\bar{i},\bar{j}} = g^{\sigma'_{\bar{i},\bar{j}}}(g^b)^{-r'_{\bar{i}}c'_{\bar{j}}/\theta},\ K''_{\bar{i},\bar{j}} = (K'_{\bar{i},\bar{j}})^{z_{\bar{i}}},$$
$$\{\bar{K}_{\bar{i},\bar{j},j'} = \big(K'_{\bar{i},\bar{j}}\big)^{\theta_{j'}}\}_{j'\in[m]\setminus\{\bar{j}\}},$$

for $k \in [l]\ s.t.\ \rho(k) \in S^*$,

$$K_{i,j,k,1} = f^{(A_k\cdot\boldsymbol{u})}G^{\xi_k} = f^{(A_k\cdot\boldsymbol{u}')+\sigma_{\bar{i},\bar{j}}(A_k\cdot\bar{\boldsymbol{u}})}G^{\xi_k} = f^{(A_k\cdot\boldsymbol{u}')}G^{\xi_k},\quad K_{i,j,k,2} = (H^{\rho(k)}h)^{-\xi_k},\quad K_{i,j,k,3} = g^{\xi_k},$$

for $k \in [l]\ s.t.\ \rho(k) \notin S^*$,

$$K_{i,j,k,1} = f^{(A_k\cdot\boldsymbol{u})}G^{\xi_k}$$
$$= f^{(A_k\cdot\boldsymbol{u}')} \cdot f^{(\sigma'_{\bar{i},\bar{j}} - br'_{\bar{i}}c'_{\bar{j}}/\theta)(A_k\cdot\bar{\boldsymbol{u}})} \cdot G^{\xi'_k} \cdot \big(g^{a\delta}\big)^{br'_{\bar{i}}c'_{\bar{j}}(A_k\cdot\bar{\boldsymbol{u}})/\delta} \cdot \big(g^{a\delta}\big)^{-r'_{\bar{i}}c'_{\bar{j}}\sum_{t\in[l^*]}\frac{acdd_t(A_k\cdot\bar{\boldsymbol{u}})/\delta}{\rho(k)-a_t^*}}$$

$$= f^{(A_k\cdot \boldsymbol{u}')} \cdot f^{\sigma'_{\bar{i},\bar{j}}(A_k\cdot\bar{\boldsymbol{u}})} \cdot G_k^{\xi'} \cdot \Big(\prod_{t\in[l^*]} (g^{a^2cdd_t})^{-r'_{\bar{i}}c'_{\bar{j}}\frac{(A_k\cdot\bar{\boldsymbol{u}})}{\rho(k)-a_t^*}} \Big),$$

$$K_{i,j,k,2} = (H^{\rho(k)}h)^{-\xi_k}$$

$$= (H^{\rho(k)}h)^{-\xi'_k} \cdot \Big(g^{\eta\rho(k)+\beta} \cdot \Big(\prod_{t\in[l^*]} (g^{b/d_t^2})^{\rho(k)-a_t^*} \Big) \cdot \Big(\prod_{t\in[l^*]} g^{acd/d_t} \Big) \Big)^{-br'_{\bar{i}}c'_{\bar{j}}(A_k\cdot\bar{\boldsymbol{u}})/\delta + r'_{\bar{i}}c'_{\bar{j}}\sum_{t'\in[l^*]}\frac{acdd_{t'}(A_k\cdot\bar{\boldsymbol{u}})/\delta}{\rho(k)-a_{t'}^*}}$$

$$= \underbrace{(H^{\rho(k)}h)^{-\xi'_k} \cdot \big(g^b\big)^{-(\eta\rho(k)+\beta)r'_{\bar{i}}c'_{\bar{j}}(A_k\cdot\bar{\boldsymbol{u}})/\delta} \cdot \Big(\prod_{t'\in[l^*]} (g^{acdd_{t'}})^{r'_{\bar{i}}c'_{\bar{j}}\frac{(\eta\rho(k)+\beta)(A_k\cdot\bar{\boldsymbol{u}})/\delta}{\rho(k)-a_{t'}^*}} \Big)}_{\Psi_1}$$

$$\cdot \underbrace{\Big(\prod_{t\in[l^*]} (g^{b^2/d_t^2})^{-(\rho(k)-a_t^*)r'_{\bar{i}}c'_{\bar{j}}(A_k\cdot\bar{\boldsymbol{u}})/\delta} \Big)}_{\Psi_2} \cdot \Big(\prod_{t\in[l^*]}\prod_{t'\in[l^*]} (g^{abcdd_{t'}/d_t^2})^{r'_{\bar{i}}c'_{\bar{j}}\frac{(\rho(k)-a_t^*)(A_k\cdot\bar{\boldsymbol{u}})/\delta}{\rho(k)-a_{t'}^*}} \Big)$$

$$\cdot \Big(\prod_{t\in[l^*]} (g^{abcd/d_t})^{-r'_{\bar{i}}c'_{\bar{j}}(A_k\cdot\bar{\boldsymbol{u}})/\delta} \Big) \cdot \Big(\prod_{t\in[l^*]}\prod_{t'\in[l^*]} (g^{a^2c^2d^2d_{t'}/d_t})^{r'_{\bar{i}}c'_{\bar{j}}\frac{(A_k\cdot\bar{\boldsymbol{u}})/\delta}{\rho(k)-a_{t'}^*}} \Big)$$

$$= \Psi_1\cdot\Psi_2 \cdot \underbrace{\Big(\prod_{t\in[l^*]}\prod_{t'\in[l^*]\setminus\{t\}} (g^{abcdd_{t'}/d_t^2})^{r'_{\bar{i}}c'_{\bar{j}}\frac{(\rho(k)-a_t^*)(A_k\cdot\bar{\boldsymbol{u}})/\delta}{\rho(k)-a_{t'}^*}} \Big)}_{\Psi_3 \quad (for\ t'\neq t)} \cdot \underbrace{\Big(\prod_{t\in[l^*]} (g^{abcdd_t/d_t^2})^{r'_{\bar{i}}c'_{\bar{j}}\frac{(\rho(k)-a_t^*)(A_k\cdot\bar{\boldsymbol{u}})/\delta}{\rho(k)-a_t^*}} \Big)}_{\Delta \quad (for\ t'=t)}$$

$$\cdot \underbrace{\Big(\prod_{t\in[l^*]} (g^{abcd/d_t})^{-r'_{\bar{i}}c'_{\bar{j}}(A_k\cdot\bar{\boldsymbol{u}})/\delta} \Big)}_{\Delta^{-1}} \cdot \underbrace{\Big(\prod_{t\in[l^*]}\prod_{t'\in[l^*]\setminus\{t\}} (g^{a^2c^2d^2d_{t'}/d_t})^{r'_{\bar{i}}c'_{\bar{j}}\frac{(A_k\cdot\bar{\boldsymbol{u}})/\delta}{\rho(k)-a_{t'}^*}} \Big)}_{\Psi_4 \quad (for\ t'\neq t)} \cdot \underbrace{\Big(\prod_{t\in[l^*]} (g^{a^2c^2d^2d_t/d_t})^{r'_{\bar{i}}c'_{\bar{j}}\frac{(A_k\cdot\bar{\boldsymbol{u}})/\delta}{\rho(k)-a_t^*}} \Big)}_{for\ t'=t}$$

$$= \Psi_1\cdot\Psi_2\cdot\Psi_3\cdot\Psi_4 \cdot \Big(\prod_{t\in[l^*]} (g^{a^2c^2d^2})^{r'_{\bar{i}}c'_{\bar{j}}\frac{(A_k\cdot\bar{\boldsymbol{u}})/\delta}{\rho(k)-a_t^*}} \Big),$$

$$K_{i,j,k,3} = g^{\xi_k} = g^{\xi'_k + br'_{\bar{i}}c'_{\bar{j}}(A_k\cdot\bar{\boldsymbol{u}})/\delta - r'_{\bar{i}}c'_{\bar{j}}\sum_{t\in[l^*]}\frac{acdd_t(A_k\cdot\bar{\boldsymbol{u}})/\delta}{\rho(k)-a_t^*}} = g^{\xi'_k}\cdot(g^b)^{r'_{\bar{i}}c'_{\bar{j}}(A_k\cdot\bar{\boldsymbol{u}})/\delta} \cdot \Big(\prod_{t\in[l^*]} (g^{acdd_t})^{-r'_{\bar{i}}c'_{\bar{j}}\frac{(A_k\cdot\bar{\boldsymbol{u}})/\delta}{\rho(k)-a_t^*}} \Big).$$

Note that $\mathcal{B}$ can calculate the values of $K_{i,j}, K'_{i,j}, K''_{i,j}, \{\bar{K}_{\bar{i},\bar{j},j'}\}_{j'\in[m]\setminus\{\bar{j}\}}, \{K_{i,j,k,1}, K_{i,j,k,2}, K_{i,j,k,3}\}_{k\in[l]}$ using the suitable terms of the assumption.

Challenge. $\mathcal{A}$ submits a message M and a revocation list R^*. $\mathcal{B}$ randomly chooses

$$\begin{aligned} \tau',\ s_1,\ldots,s_{\bar{i}-1},s'_{\bar{i}},s_{\bar{i}+1},\ldots,s_m,\ t'_1,\ldots,t'_{\bar{i}-1},t_{\bar{i}},t'_{\bar{i}+1},\ldots,t'_m &\in \mathbb{Z}_p, \\ \boldsymbol{w}_1,\ldots,\boldsymbol{w}_{\bar{j}-1},\boldsymbol{w}'_{\bar{j}},\ldots,\boldsymbol{w}'_m &\in \mathbb{Z}_p^3, \\ \pi',\ \delta'_1,\ldots,\delta'_{l^*} &\in \mathbb{Z}_p. \end{aligned}$$

$\mathcal{B}$ randomly chooses $r_x, r_y, r_z \in \mathbb{Z}_p$, and sets $\boldsymbol{\chi}_1 = (r_x, 0, r_z), \boldsymbol{\chi}_2 = (0, r_y, r_z), \boldsymbol{\chi}_3 = \boldsymbol{\chi}_1 \times \boldsymbol{\chi}_2 = (-r_y r_z, -r_x r_z, r_x r_y)$. $\mathcal{B}$ randomly chooses

$$\begin{gathered} \boldsymbol{v}_i \in \mathbb{Z}_p^3\ \forall i \in \{1,\ldots,\bar{i}-1\}, \\ \boldsymbol{v}_{\bar{i}}^p \in span\{\boldsymbol{\chi}_1,\boldsymbol{\chi}_2\},\ \boldsymbol{v}_{\bar{i}}^q \in span\{\boldsymbol{\chi}_3\}, \end{gathered}$$

$$\boldsymbol{v}_i \in span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}\ \forall i \in \{\bar{i}+1, \ldots, m\},$$
$$\boldsymbol{v}_c^p \in span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\},\ \boldsymbol{v}_c^q = \nu_3 \boldsymbol{\chi}_3 \in span\{\boldsymbol{\chi}_3\}.$$

$\mathcal{B}$ sets the value of $\kappa, \tau, s_{\bar{i}}, t_i (i \in [m] \setminus \{\bar{i}\}) \in \mathbb{Z}_p$, $\boldsymbol{v}_c, \boldsymbol{v}_{\bar{i}} \in \mathbb{Z}_p^3$, $\{\boldsymbol{w}_j \in \mathbb{Z}_p^3\}_{j=\bar{j}}^m$, $\pi \in \mathbb{Z}_p, \{\delta_t \in \mathbb{Z}_p\}_{t\in[l^*]}$ by implicitly setting

$$a \equiv \kappa \bmod p,\quad ac\tau' \equiv \tau \bmod p,\quad s'_{\bar{i}}/a \equiv s_{\bar{i}} \bmod p,$$
$$t'_i + cd\theta\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)/z'_i \equiv t_i \bmod p\ \forall i \in \{1, \ldots, \bar{i}-1\},$$
$$t'_i - a\theta\tau' s_i(\boldsymbol{v}_i \cdot \boldsymbol{v}_c^p)/z'_i + cd\theta\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)/z'_i \equiv t_i \bmod p\ \forall i \in \{\bar{i}+1, \ldots, m\},$$
$$\boldsymbol{v}_c = c^{-1}\boldsymbol{v}_c^p + \boldsymbol{v}_c^q,\ \ \boldsymbol{v}_{\bar{i}} = \boldsymbol{v}_{\bar{i}}^p + d\boldsymbol{v}_{\bar{i}}^q,$$
$$\boldsymbol{w}'_{\bar{j}} - bc'_{\bar{j}}\tau'\boldsymbol{v}_c^p \equiv \boldsymbol{w}_{\bar{j}} \bmod p,$$
$$\boldsymbol{w}'_j - cdc_j\tau'\boldsymbol{v}_c^q \equiv \boldsymbol{w}_j \bmod p\ \forall j \in \{\bar{j}+1, \ldots, m\},$$
$$\pi' - cd\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q) \equiv \pi \bmod p,\quad \delta'_t - d_t\delta\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q) \equiv \delta_t \bmod p\ \forall t \in [l^*].$$

It is worth noticing that $\boldsymbol{v}_{\bar{i}}$ and $\boldsymbol{v}_c$ are random vectors in $\mathbb{Z}_p^3$ as required, and $(\boldsymbol{v}_{\bar{i}} \cdot \boldsymbol{v}_c) = \frac{1}{c}(\boldsymbol{v}_{\bar{i}}^p \cdot \boldsymbol{v}_c^p) + d(\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)$, since $\boldsymbol{\chi}_3$ is orthogonal to $span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2\}$ and $\mathbb{Z}_p^3 = span\{\boldsymbol{\chi}_1, \boldsymbol{\chi}_2, \boldsymbol{\chi}_3\}$.

Let $\bar{R}^* = [m, m] \backslash R^*$ and $\bar{R}_i^* = \{j' | (i, j') \in \bar{R}^*\}\ \forall i \in [m]$. $\mathcal{B}$ creates the ciphertext $\langle R^*, S^*, (\boldsymbol{R}_i, \boldsymbol{R}'_i, Q_i, Q'_i, Q''_i, T_i)_{i=1}^m, (\boldsymbol{C}_j, \boldsymbol{C}'_j)_{j=1}^m, P, \{P_x, P'_x\}_{x\in S^*}\rangle$ as follows:

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

$$\boldsymbol{R}_i = g^{\boldsymbol{v}_i},\ \boldsymbol{R}'_i = (g^a)^{\boldsymbol{v}_i},$$
$$Q_i = g^{s_i},\ Q'_i = (f \prod_{j'\in\bar{R}_i^*} f_{j'})^{s_i} Z_i^{t'_i} f^{\pi'},\ Q''_i = g^{t'_i}(g^{cd})^{\theta\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)/z'_i},\ \ T_i = E_i^{\hat{s}_i}.$$

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

$$\boldsymbol{R}_{\boldsymbol{i}} = g^{r'_{\bar{i}} s'_{\bar{i}} \boldsymbol{v}_{\bar{i}}^p}(g^d)^{r'_{\bar{i}} s'_{\bar{i}} \boldsymbol{v}_{\bar{i}}^q},\ \ \boldsymbol{R}'_i = (g^a)^{r'_{\bar{i}} s'_{\bar{i}} \boldsymbol{v}_{\bar{i}}^p}(g^{ad})^{r'_{\bar{i}} s'_{\bar{i}} \boldsymbol{v}_{\bar{i}}^q},$$
$$Q_i = g^{\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}^p \cdot \boldsymbol{v}_c^p)}(g^{cd})^{\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)},\ Q'_i = f^{\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}^p \cdot \boldsymbol{v}_c^p)}(\prod_{j'\in\bar{R}_i^*} Q_i^{\theta_{j'}}) Z_i^{t_{\bar{i}}} f^{\pi'},\ Q''_i = g^{t_{\bar{i}}},$$
$$T_i = M \cdot e(g^{\alpha_i}, Q_i).$$

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

$$\boldsymbol{R}_{\boldsymbol{i}} = g^{r_i s_i \boldsymbol{v}_i},\ \ \boldsymbol{R}'_i = (g^a)^{r_i s_i \boldsymbol{v}_i},$$
$$Q_i = (g^a)^{\tau' s_i(\boldsymbol{v}_i \cdot \boldsymbol{v}_c^p)},\ Q'_i = (\prod_{j'\in\bar{R}_i^*} Q_i^{\theta_{j'}}) Z_i^{t'_i} f^{\pi'},\ Q''_i = g^{t'_i}(g^a)^{-\theta\tau' s_i(\boldsymbol{v}_i \cdot \boldsymbol{v}_c^p)/z'_i}(g^{cd})^{\theta\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)/z'_i},$$
$$T_i = M \cdot e(g^{\alpha_i}, Q_i).$$

2. For each $j \in [m]$:
   - if $j < \bar{j}$: it randomly chooses $\mu'_j \in \mathbb{Z}_p$ and implicitly sets the value of μ_j such that $(acd)^{-1}\mu'_j\nu_3 - \nu_3 \equiv \mu_j \bmod p$, then sets $\boldsymbol{C}_j = (g^{ad})^{c'_j\tau'\boldsymbol{v}_c^p} \cdot g^{c'_j\tau'\mu'_j\boldsymbol{v}_c^q} \cdot (g^a)^{\boldsymbol{w}_j},\ \ \boldsymbol{C}'_j = g^{\boldsymbol{w}_j}.$

– if $j = \bar{j}$: it sets $\boldsymbol{C}_j = T^{c'_{\bar{j}}\tau'\boldsymbol{v}_c^q} \cdot (g^a)^{\boldsymbol{w}'_j}$, $\boldsymbol{C}'_j = g^{\boldsymbol{w}'_{\bar{j}}} \cdot (g^b)^{-c'_{\bar{j}}\tau'\boldsymbol{v}_c^p}$.

– if $j > \bar{j}$: it sets $\boldsymbol{C}_j = (g^{ad})^{c'_j\tau'\boldsymbol{v}_c^p} \cdot (g^a)^{\boldsymbol{w}'_j}$, $\boldsymbol{C}'_j = g^{\boldsymbol{w}'_j} \cdot (g^{cd})^{-c'_j\tau'\boldsymbol{v}_c^q}$.

3. $P = g^\pi = g^{\pi'}(g^{cd})^{-\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)}$, and for $t \in [l^*]$,

$$
\begin{aligned}
&P_{a_t^*} = g^{\delta_t} = g^{\delta'_t}(g^{d_t})^{-\delta\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)},\\
&P'_{a_t^*} = (H^{a_t^*}h)^{\delta_t} G^{-\pi}\\
&= \underbrace{(H^{a_t^*}h)^{\delta'_t}}_{\Phi_1} \cdot \Big(g^{\eta a_t^* + \beta} \cdot \big(\prod_{t' \in [l^*]} (g^{b/d_{t'}^2})^{a_t^* - a_{t'}^*} \big) \cdot \big(\prod_{t' \in [l^*]} g^{acd/d_{t'}} \big) \Big)^{-d_t \delta\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)}\\
&\quad \cdot G^{-\pi'} \cdot (g^{a\delta})^{cd\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)}\\
&= \Phi_1 \cdot \underbrace{\big(g^{d_t}\big)^{-(\eta a_t^* + \beta)\delta\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)}}_{\Phi_2} \cdot \big(\prod_{t' \in [l^*]} (g^{bd_t/d_{t'}^2})^{-(a_t^* - a_{t'}^*)\delta\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)} \big) \cdot \big(\prod_{t' \in [l^*]} (g^{acdd_t/d_{t'}})^{-\delta\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)} \big)\\
&\quad \cdot G^{-\pi'} \cdot (g^{acd})^{\delta\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)}\\
&= \Phi_1 \cdot \Phi_2 \cdot \underbrace{\big(\prod_{t' \in [l^*] \setminus \{t\}} (g^{bd_t/d_{t'}^2})^{-(a_t^* - a_{t'}^*)\delta\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)} \big)}_{\Phi_3 \quad (for\ t' \neq t)} \cdot \underbrace{\big((g^{bd_t/d_t^2})^{-(a_t^* - a_t^*)\delta\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)}\big)}_{1 \quad (for\ t' = t)}\\
&\quad \cdot \underbrace{\big(\prod_{t' \in [l^*] \setminus \{t\}} (g^{acdd_t/d_{t'}})^{-\delta\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)} \big)}_{\Phi_4 \quad (for\ t' \neq t)} \cdot \underbrace{\big((g^{acdd_t/d_t})^{-\delta\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)}\big)}_{\Delta \quad (for\ t' = t)} \cdot G^{-\pi'} \cdot \underbrace{(g^{acd})^{\delta\tau' s'_{\bar{i}}(\boldsymbol{v}_{\bar{i}}^q \cdot \boldsymbol{v}_c^q)}}_{\Delta^{-1}}\\
&= \Phi_1 \cdot \Phi_2 \cdot \Phi_3 \cdot \Phi_4 \cdot G^{-\pi'}.
\end{aligned}
$$

Note that the values of $\Phi_1, \ldots, \Phi_4$ can be calculated using the suitable terms of the assumption.

If $T = g^{abc}$, then the ciphertext is a well-formed encryption to the index $(\bar{i}, \bar{j})$. If T is randomly chosen, say $T = g^r$ for some random $r \in \mathbb{Z}_p$, the ciphertext is a well-formed encryption to the index $(\bar{i}, \bar{j} + 1)$ with implicitly setting $\mu_{\bar{j}}$ such that $(\frac{r}{abc} - 1)\nu_3 \equiv \mu_{\bar{j}} \bmod p$.

Phase 2. Same as **Phase 1.**

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

When $\mathcal{B}$ does not abort, the distributions of the public parameter, private keys and challenge ciphertext are the same as in the real scheme, $\mathcal{B}$'s advantage in Source Group q-parallel DBDH game will be exactly equal to $\mathcal{A}$'s advantage in the selective index-hiding game. Thus, $\mathcal{B}$'s final advantage is $Adv_{\mathcal{B},q} = Adv_{\mathcal{A}} \cdot \Pr[\mathcal{A} \text{ is not in } \textbf{Case II.1} \wedge (c = 1)]$.

Note that in both **Case A** and **Case B**, the distributions of the public parameter, private keys and challenge ciphertext that $\mathcal{B}$ gives $\mathcal{A}$ are the same as in the real scheme and independent of the value of c. This implies that the value of c and the case that $\mathcal{A}$ is in are independent of each other. Let $\mathcal{A}.\textbf{I}, \mathcal{A}.\textbf{II.1}, \mathcal{A}.\textbf{II.2}$, and $\mathcal{A}.\textbf{II.3}$ be the events that $\mathcal{A}$ is in **Case I**, **Case II.1**, **Case II.2** and **Case II.3**, respectively, and $\overline{\mathcal{A}.\textbf{II.1}}$ and $\overline{\mathcal{A}.\textbf{II.3}}$ be the events that "$\mathcal{A}$ is not in **Case II.1**" and "$\mathcal{A}$ is not in **Case II.3**", respectively. We have

$$Adv_{\mathcal{B},3} + Adv_{\mathcal{B},q} = Adv_{\mathcal{A}} \cdot \Pr[\overline{\mathcal{A}.\textbf{II.3}} \wedge (c = 0)] + Adv_{\mathcal{A}} \cdot \Pr[\overline{\mathcal{A}.\textbf{II.1}} \wedge (c = 1)]$$

$$
\begin{aligned}
&= Adv_{\mathcal{A}} \cdot \Pr[\overline{\mathcal{A}.\mathbf{II}.\mathbf{3}}] \cdot \Pr[c = 0] + Adv_{\mathcal{A}} \cdot \Pr[\overline{\mathcal{A}.\mathbf{II}.\mathbf{1}}] \cdot \Pr[c = 1] \\
&= Adv_{\mathcal{A}} \cdot (1 - \Pr[\mathcal{A}.\mathbf{II}.\mathbf{3}]) \cdot \frac{1}{2} + Adv_{\mathcal{A}} \cdot (1 - \Pr[\mathcal{A}.\mathbf{II}.\mathbf{1}]) \cdot \frac{1}{2} \\
&= \frac{1}{2} \cdot Adv_{\mathcal{A}} \cdot (2 - (\Pr[\mathcal{A}.\mathbf{II}.\mathbf{3}] + \Pr[\mathcal{A}.\mathbf{II}.\mathbf{1}])) \\
&\geq \frac{1}{2} \cdot Adv_{\mathcal{A}},
\end{aligned}
$$

since $\Pr[\mathcal{A}.\mathbf{II}.\mathbf{3}] + \Pr[\mathcal{A}.\mathbf{II}.\mathbf{1}] \leq \Pr[\mathcal{A}.\mathbf{I}] + \Pr[\mathcal{A}.\mathbf{II}.\mathbf{1}] + \Pr[\mathcal{A}.\mathbf{II}.\mathbf{2}] + \Pr[\mathcal{A}.\mathbf{II}.\mathbf{3}] = 1$. This implies that either $Adv_{\mathcal{B},3} \geq \frac{1}{4} \cdot Adv_{\mathcal{A}}$ or $Adv_{\mathcal{B},q} \geq \frac{1}{4} \cdot Adv_{\mathcal{A}}$.

J Proof of Our Source Group Parallel DBDH Assumption

In this section, we give a lower bound to the complexity of our Source Group q-Parallel DBDH Assumption. The proof is similar to that of the Source Group q-Parallel BDHE Assumption [19], which is given in [19, Appendix B] in the generic group model. In the generic group model [29], an adversary does not have direct access to the group. It must interact with an oracle to perform the group operation and obtain "handles" for new elements. Also, it can only use handles previously received from the oracle. We consider an experiment where an adversary is given handles for the group elements given out in the assumption as well as a handle for the challenge term T_β (here, β is a uniformly random bit). The adversary may interact with the oracle to perform group operations and pairings, and gets handles in return as the results from these operations. Finally, the adversary guesses the bit β. The difference between the adversary's success probability and one half is defined as its advantage. We refer readers to [3,13] for other examples of using the generic group model for justifying assumptions in bilinear groups. We denote $a, b, c, d, b_1, \ldots, b_q$ as variables over $\mathbb{Z}_p$, and define $\mathcal{M}$ as the following set of rational functions over these variables:

$$
\begin{aligned}
\mathcal{M} := \{\ & 1, a, b, cd, d, ad, (acd)^2 \\
& d_i,\ acdd_i,\ a^2cdd_i,\ acd/d_i,\ b/d_i^2,\ b^2/d_i^2 \qquad \forall i \in [q], \\
& acdd_i/d_j,\ (acd)^2 d_i/d_j,\ bd_i/d_j^2,\ abcdd_i/d_j^2 \ \ \forall i, j \in [q]\ s.t.\ i \neq j\ \}
\end{aligned}
$$

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

Lemma 6. *For each function $M \in \mathcal{M} \cup \{abc\}$, the product $M \cdot abc$ is independent of $E(\mathcal{M}) \cup abc(\mathcal{M} \setminus M)$. (Here, $abc(\mathcal{M} \setminus M)$ denotes the set formed by removing M from $\mathcal{M}$ and then multiplying all remaining elements by abc.)*

Proof. As every element in $\mathcal{M} \cup \{abc\}$ and $E(\mathcal{M}) \cup abc(\mathcal{M} \setminus M)$ is a ratio of monomials, the only way that $M \cdot abc$ can be dependent on $E(\mathcal{M}) \cup abc(\mathcal{M} \setminus M)$ is if it is *contained* in $E(\mathcal{M}) \cup abc(\mathcal{M} \setminus M)$. First, $(abc)^2$ is not in $E(\mathcal{M}) \cup abc\mathcal{M}$, and for any $M \in \mathcal{M}$, $abcM \notin abc(\mathcal{M} \setminus M)$. Thus it suffices to show that for any M, $abcM \notin E(\mathcal{M})$. In other words, we show that $E(\mathcal{M})$ does not intersect with

the set $abc\mathcal{M}$, which is formed by multiplying each element of $\mathcal{M}$ by abc. To see this, we examine the set $abc\mathcal{M}$. By definition, we have that

$$\begin{aligned} abc\mathcal{M} := \{\ & abc, a^2bc, ab^2c, abc^2d, abcd, a^2bcd, a^3bc^3d^2 \\ & abcd_i,\ a^2bc^2dd_i,\ a^3bc^2dd_i,\ a^2bc^2d/d_i,\ ab^2c/d_i^2,\ ab^3c/d_i^2 \quad \forall i \in [q], \\ & a^2bc^2dd_i/d_j,\ a^3bc^3d^2d_i/d_j,\ ab^2cd_i/d_j^2,\ a^2b^2c^2dd_i/d_j^2 \qquad \forall i,j \in [q]\ s.t.\ i \neq j\ \} \end{aligned}$$

We now check if any of these are in $E(\mathcal{M})$, which is the set of pairwise products of things in $\mathcal{M}$. In $\mathcal{M}$, every occurrence of c is accompanied by d, and d^{-1} never appears. Hence $E(\mathcal{M})$ does not contain any element which has a higher powers of c than d. This rules out all the elements in $abc\mathcal{M}$ above but $abcd$ and a^2bcd. To rule out $abcd$, we consider all the possible ways it might be formed as a product of two elements of $\mathcal{M}$. As d is in the term, one of the two factors in $\mathcal{M}$ must be a term containing d. At the same time, as $a^{-1}, b^{-1}, c^{-1}, d^{-1}$ never appear, if any one of $\{a, b, c, d\}$ has order ≥ 2 then the term could not be a factor. Note that d, cd or ad cannot be one of the factors as $abc, ab, bc \notin \mathcal{M}$. Also, an element of the form $acdd_i$ cannot be one of the two factors as $b/d_i \notin \mathcal{M}$, an element of the form acd/d_i cannot be one of the two factors as $bd_i \notin \mathcal{M}$, an element of the form $acdd_i/d_j$ $(s.t.\ j \neq j')$ cannot be one of the two factors as $bd_j/d_i \notin \mathcal{M}$, and an element of the form $abcdd_i/d_j^2$ $(s.t.\ j \neq j')$ cannot be one of the two factors as $d_j^2/d_i \notin \mathcal{M}$. Hence we can dismiss all the possible ways, and conclude that $abcd \notin E(\mathcal{M})$. To rule out a^2bcd, we consider all the possible ways it might be formed as a product of two elements of $\mathcal{M}$. As d is in the term, one of the two factors in $\mathcal{M}$ must be a term containing d. At the same time, as $a^{-1}, b^{-1}, c^{-1}, d^{-1}$ never appear, if any one of $\{b, c, d\}$ has order ≥ 2 or a has order ≥ 3 then the term could not be a factor. Note that d, cd or ad cannot be one of the factors as $a^2bc, a^2b, abc \notin \mathcal{M}$. Also, an element of the form $acdd_i$ cannot be one of the two factors as $ab/d_i \notin \mathcal{M}$, an element of the form a^2cdd_i cannot be one of the two factors as $b/d_i \notin \mathcal{M}$, an element of the form acd/d_i cannot be one of the two factors as $abd_i \notin \mathcal{M}$, an element of the form $acdd_i/d_j$ $(s.t.\ j \neq j')$ cannot be one of the two factors as $abd_j/d_i \notin \mathcal{M}$, and an element of the form $abcdd_i/d_j^2$ $(s.t.\ j \neq j')$ cannot be one of the two factors as $ad_j^2/d_i \notin \mathcal{M}$. Hence we can dismiss all the possible ways, and conclude that $abcd \notin E(\mathcal{M})$.

We now proceed similarly to the proof strategy in [3,13] to establish the following theorem:

Theorem 10. *For any adversary $\mathcal{A}$ that makes Q queries to the oracles computing the group operations in $\mathbb{G}, \mathbb{G}_T$ and the bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, the advantages of $\mathcal{A}$ against the Source Group q-Parallel DBDH assumption in the generic group model is at most $O(\frac{Q^2q}{p})$.*

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