A Framework and Compact Constructions for Non-monotonic Attribute-Based Encryption

Shota Yamada* Nuttapong Attrapadung† Goichiro Hanaoka† Noboru Kunihiro*

Abstract

In this paper, we propose new non-monotonic attribute-based encryption schemes with compact parameters. The first three schemes are key-policy attribute-based encryption (KP-ABE) and the fourth scheme is ciphertext-policy attribute-based encryption (CP-ABE) scheme.

- Our first scheme has very compact ciphertexts. The ciphertext overhead only consists of two group elements and this is the shortest in the literature. Compared to the scheme by Attrapadung et al. (PKC2011), which is the best scheme in terms of the ciphertext overhead, our scheme shortens ciphertext overhead by 33%. The scheme also reduces the size of the master public key to about half.
- Our second scheme is proven secure under the decisional bilinear Diffie-Hellman (DBDH) assumption, which is one of the most standard assumptions in bilinear groups. Compared to the non-monotonic KP-ABE scheme from the same assumption by Ostrovsky et al. (ACM-CCS'07), our scheme achieves more compact parameters. The master public key and the ciphertext size is about the half that of their scheme.
- Our third scheme is the first non-monotonic KP-ABE scheme that can deal with unbounded size of set and access policies. That is, there is no restriction on the size of attribute sets and the number of allowed repetition of the same attributes which appear in an access policy. The master public key of our scheme is very compact: it consists of only constant number of group elements.
- Our fourth scheme is the first non-monotonic CP-ABE scheme that can deal with unbounded size of set and access policies. The master public key of the scheme consists of only constant number of group elements.

We construct our KP-ABE schemes in a modular manner. We first introduce special type of predicate encryption that we call two-mode identity based broadcast encryption (TIBBE). Then, we show that any TIBBE scheme that satisfies certain condition can be generically converted into non-monotonic KP-ABE scheme. Finally, we construct efficient TIBBE schemes and apply this conversion to obtain the above new non-monotonic KP-ABE schemes.

Keywords. Attribute-based encryption, non-monotonic access structure, compact parameters.

*The University of Tokyo. The first author is supported by Research Fellowship for Young Scientists.
†National Institute of Advanced Industrial Science and Technology (AIST).

1 Introduction

In many systems, a server monitors access to sensitive data so that only certain users can access it. If the server is not fully trusted, the data must be encrypted. However, a standard public key encryption scheme is not appropriate, because it severely limits the users who can access the contents.

To solve this problem, Sahai and Waters [31] were the first to study attribute-based encryption (ABE). In ABE, one can encrypt data for a set of receivers that satisfy certain condition. In Sahai and Waters' scheme, a ciphertext and a private key are associated with a set of attributes, and the key can decrypt the ciphertext if and only if these sets overlap more than certain threshold. Goyal, Pandey, Sahai, and Waters [16] further extended their result and proposed schemes that support finer-grained access control. In their scheme, a ciphertext is associated with a set of attributes, and a private key is associated with an access structure that is specified by a Boolean formula. Decryption is possible when the set satisfies this Boolean formula. Their schemes are called key-policy ABE (KP-ABE), because the key specifies the access structure. Ciphertext-policy ABE (CP-ABE) is complementary form to KP-ABE in the sense that a ciphertext specifies an access structure while a key is associated with a set of attributes. The first studies of CP-ABE appear in [5, 12].

The above schemes can express a wide class of access structures, but they are still limited because they only support a monotonic access structure. In particular, they cannot deal with an access structure that is associated with a Boolean formula that includes the negation of attributes. This is not convenient for real world applications. One possible solution to this problem is to explicitly include attributes that express absence of attributes in the attribute space, as suggested in [16]. For example, in the KP-ABE case, to encrypt a message for an attribute x_1, one should encrypt the message for a set that includes x_1 and attributes "Not x_j" for all attribute x_j such that $x_j \neq x_1$, using the underlying monotonic KP-ABE system. Then, a key with the attribute "Not x_2" can decrypt the ciphertext and recover the message as desired, because "Not x_2" $\in \{x_1$, "Not x_2", "Not x_3", $\ldots, \}$. This solution is not appropriate for many applications, because the scheme becomes inefficient as the total number of attributes increase. Furthermore, it does not work in the setting where attribute space is exponentially large. Ostrovsky, Sahai, and Waters [28] addressed this problem and constructed the first KP-ABE scheme that supports a non-monotonic access structure by using an idea from the Naor-Pinkas revocation scheme [25]. Following their work, several non-monotonic KP/CP-ABE schemes have been proposed [21, 26, 3, 27].

Our Contributions. In this paper, we propose new non-monotonic ABE schemes. Our new schemes either improve efficiency or achieve a new functionality that was previously not possible. We propose the following four schemes. The first three schemes are KP-ABE schemes and the last one is CP-ABE scheme.

- The first scheme has very compact ciphertexts. The ciphertext overhead of our scheme consists of only two group elements, which is even shorter than the currently shortest scheme of [3]. Furthermore, the scheme also reduces the size of master public key to about half while the private key size is slightly larger.
- The second scheme is proven secure under the decisional bilinear Diffie-Hellman (DBDH) assumption, which is one of the weakest number theoretic assumptions in bilinear groups. The public key and the ciphertext size of our scheme are about half the size of the scheme in [28], which is secure under the same assumption. The encryption algorithm of our scheme is at least two times faster than the existing scheme, but our decryption algorithm is somewhat slower.
- The third scheme is the first non-monotonic KP-ABE scheme in the standard model that supports fully unbounded attribute sets and access policies. That is, there is no restriction on the size of the attribute set, or on the number of times the same attributes can appear in an access policy. The master public key of the scheme is very compact: it consists of only

constant number of group elements. Such a construction has previously only been possible in the random oracle model [21]. *

- The fourth scheme is the first non-monotonic CP-ABE scheme that supports fully unbounded size of attribute sets and access policies. The master public key of our scheme consists of only constant number of group elements.

We construct the above KP-ABE schemes in a modular way. First, we define a new predicate encryption that we call two mode identity based broadcast encryption (TIBBE). In TIBBE, a ciphertext is associated with a set of identities. A private key is associated with an identity and certain "type". There are two types of keys in the system. First type keys can decrypt the ciphertext iff the identity is included in the set, while the second type keys can iff the identity is *not* included. The notion of TIBBE is an extension of identity based broadcast encryption (IBBE) and identity based revocation (IBR). We show that any TIBBE scheme with a certain property can be generically converted into a non-monotonic KP-ABE scheme. This can be seen as an extension of the previous result in [3] that converts any IBBE scheme with certain properties into a (monotonic) KP-ABE scheme. Finally, we construct efficient TIBBE schemes. By applying our conversion to these schemes, we obtain our new non-monotonic KP-ABE schemes.

While we construct KP-ABE schemes in a modular way, our construction of the above non-monotonic CP-ABE scheme is more direct. Our construction is based on the (monotonic) CP-ABE scheme recently proposed by [30]. We extend their scheme to support a non-monotonic access structure by applying an idea from the IBR scheme in [21] to the CP-ABE setting.

Finally, we remark that all our schemes are selectively secure. Constructing adaptively secure schemes with similar property is left open for future research.

Other Related Works. After the work of Sahai and Waters [31], many CP/KP-ABE schemes have been proposed [16, 15, 32, 17]. The first adaptively secure ABE schemes were proposed in [20] using composite order groups. Later, schemes on prime order groups were proposed [26, 27, 19, 24]. The settings with multiple-authorities are investigated in several works [10, 1, 11, 22]. To construct a scheme with even more general access structure is an important direction of research. Recently, there are significant progress toward this direction [33, 13, 14].

2 Preliminaries

2.1 Notation

We will treat a vector as a row vector, unless stated otherwise. For any vector $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{Z}_p^n$, $g^{\mathbf{a}} = (g^{a_1}, \ldots, g^{a_n})$. For $\mathbf{a}, \mathbf{z} \in \mathbb{Z}_p^n$, we denote their inner product as $\langle \mathbf{a}, \mathbf{z} \rangle = \mathbf{a} \cdot \mathbf{z}^\top = \sum_{i=1}^n a_i z_i$. We denote by $\mathbf{e}_i$ the i-th unit vector: its i-th component is one, all others are zero. We also denote by $[n]$ a set $\{1, \ldots, n\}$ for an integer $n > 0$ and $[n_1, \ldots, n_m] = [n_1] \times \cdots \times [n_m]$ for integers $n_1, \ldots, n_m > 0$. For a set U, we define $2^U = \{S | S \subseteq U\}$ and $\binom{U}{<k} = \{S | S \subseteq U, |S| < k\}$ for $k \leq |U|$.

2.2 Definition of Predicate Encryption

Here, we define the syntax of predicate encryption. We emphasize that we do not consider attribute hiding in this paper†.

Syntax. Let $R = \{R_N : A_N \times B_N \to \{0,1\} \mid N \in \mathbb{N}^c\}$ be a relation family where A_N and B_N denote "key attribute" and "ciphertext attribute" spaces and c is some fixed constant. The index

*Unbounded non-monotonic schemes were recently mentioned in [23, 30], but only monotonic constructions were given.

†This is called "public-index" predicate encryption, categorized in [9].

$N = (n_1, n_2, \ldots, n_c)$ of R_N denotes the numbers of bounds for corresponding parameters. If an index N is not required, we say that R is an unbounded relation. A predicate encryption (PE) scheme for R consists of the following algorithms:

$\mathsf{Setup}(\lambda, N) \to (\mathsf{mpk}, \mathsf{msk})$: The setup algorithm takes as input a security parameter λ and a index N of the relation R_N and outputs a master public key mpk and a master secret key msk.

$\mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, X) \to \mathsf{sk}_X$: The key generation algorithm takes as input the master secret key msk, the master public key mpk, and a key attribute $X \in A_N$. It outputs a private key sk_X. We assume X is included in sk_X implicitly.

$\mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}, Y) \to C$: The encryption algorithm takes as input a master public key mpk, the message M, and a ciphertext attribute $Y \in B_N$. It will output a ciphertext C.

$\mathsf{Decrypt}(\mathsf{mpk}, C, Y, \mathsf{sk}_X) \to \mathsf{M}$ or $\perp$: We assume that the decryption algorithm is deterministic. The decryption algorithm takes as input the public parameters mpk, a ciphertext C, ciphertext attribute $Y \in B_N$ and a private key sk_X. It outputs the message M or $\perp$ which represents that the ciphertext is not in a valid form.

We require correctness of decryption: that is, for all λ, N, all $(\mathsf{mpk}, \mathsf{msk})$ produced by $\mathsf{Setup}(\lambda, N)$, all $X \in A_N, Y \in B_N$ such that $R(X, Y) = 1$, and all sk_X returned by $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, X)$, $\mathsf{Decrypt}(\mathsf{mpk}, \mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}, Y), Y, \mathsf{sk}_X) = \mathsf{M}$ holds.

SECURITY. We now define the security for an PE scheme Π. This security notion is defined by the following game between a challenger and an attacker $\mathcal{A}$.

At first, the challenger runs the setup algorithm and gives mpk to $\mathcal{A}$. Then $\mathcal{A}$ may adaptively make key-extraction queries. We denote this phase **Phase1**. In this phase, if $\mathcal{A}$ submits X to the challenger, the challenger returns $\mathsf{sk}_X \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, X)$. At some point, $\mathcal{A}$ outputs two equal length messages M_0 and M_1 and challenge ciphertext attribute $Y^\star \in B_N$. $Y^\star$ cannot satisfy $R(X, Y^\star) = 1$ for any attribute X such that $\mathcal{A}$ already queried private key for X. Then the challenger flips a random coin $\beta \in \{0, 1\}$, runs $\mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}_\beta, Y^\star) \to C^\star$ and gives challenge ciphertext $C^\star$ to $\mathcal{A}$. In **Phase2**, $\mathcal{A}$ may adaptively make queries as in **Phase1** with following added restriction: $\mathcal{A}$ cannot make a key-extraction query for X such that $R(X, Y^\star) = 1$. At last, $\mathcal{A}$ outputs a guess β' for β. We say that $\mathcal{A}$ succeeds if $\beta' = \beta$ and denote the probability of this event by $\Pr^{PE}_{\mathcal{A},\Pi}$. The advantage of an attacker $\mathcal{A}$ is defined as $Adv^{PE}_{\mathcal{A},\Pi} = |\Pr^{PE}_{\mathcal{A},\Pi} - \frac{1}{2}|$. We say that Π is fully secure if $Adv^{PE}_{\mathcal{A},\Pi}$ is negligible for all probabilistic polynomial time (PPT) adversary $\mathcal{A}$.

A weaker notion called selective security can be defined as in the above game with the exception that the adversary $\mathcal{A}$ has to choose the challenge ciphertext index $Y^\star$ before the setup phase but private key queries $X_1, \ldots, X_q$ can still be adaptive. All schemes proposed in this paper are selectively secure.

2.3 Linear Secret Sharing Scheme and Attribute-Based Encryption

Here, we first define linear secret sharing scheme (LSSS) following [4] and then define key/ciphertext-policy atrribute based encryption scheme as an instance of PE.

Definition 1 (Access Structure). *Let $\mathcal{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_n\}$ be a set of parties. A collection $\mathbb{A} \subset 2^{\mathcal{P}}$ is said to be monotone if, for all B, C, if $B \in \mathbb{A}$ and $B \subset C$, then $C \in \mathbb{A}$ holds. An access structure (resp., monotonic access structure) is a collection (resp., monotone collection) $\mathbb{A} \subset 2^{\mathcal{P}} \backslash \{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.*

Definition 2 (Linear Secret Sharing Scheme). *Let $\mathcal{P}$ be a set of parties. Let L be an $\ell \times m$ matrix. Let $\pi : \{1, \ldots, \ell\} \to \mathcal{P}$ be a function that maps a row to a party for labeling. A secret sharing scheme*

π for access structure $\mathbb{A}$ over a set of parties $\mathcal{P}$ is a linear secret-sharing scheme (LSSS) in $\mathbb{Z}_p$ and is represented by (L, π) if it consists of two efficient algorithms:

$\mathsf{Share}_{L,\pi}$ *There exists an efficient algorithm which takes as input $s \in \mathbb{Z}_p$ which is to be shared. It chooses $s_2, \ldots, s_m \xleftarrow{\$} \mathbb{Z}_p$ and let $\mathbf{s} = (s, s_2, \ldots, s_m)$. It outputs $L \cdot \mathbf{s}$ as the vector of ℓ shares. The share $\lambda_i = \langle \mathbf{L}_i, \mathbf{s} \rangle$ belongs to party $\pi(i)$, where $\mathbf{L}_i$ denotes the i-th row of L.*

$\mathsf{Recon}_{L,\pi}$ *The algorithm takes as input an access set $S \in \mathbb{A}$. Let $I = \{i | \pi(i) \in S\}$. It outputs a set of constants $\{(i, \mu_i)\}_{i \in I}$ which has a linear reconstruction property: $\sum_{i \in I} \mu_i \cdot \lambda_i = s$.*

Terminology for Non-Monotonic Access Structure. We recall a technique by Ostrovsky Sahai, and Waters [28] to move from monotonic access structures to non-monotonic access structure. They assume a family $\{\Pi_{\mathbb{A}}\}_{\mathbb{A} \in \mathcal{AS}}$ of linear secret sharing schemes for a set of monotonic access structures $\mathbb{A}$. For each such access structure $\mathbb{A} \in \mathcal{AS}$, the set $\mathcal{P}$ of underlying parties has the following properties: The names of the parties in $\mathcal{P}$ may be of two types: either the name is normal (like x) or it is primed (like x'), and if $x \in \mathcal{P}$ then $x' \in \mathcal{P}$ and vice versa. Conceptually, prime attributes are associated with negation of unprimed attributes.

A family $\mathcal{AS}$ of non-monotone access structures can be defined as follows. For each access structure $\mathbb{A} \in \mathcal{AS}$ over a set of parties $\mathcal{P}$, one defines a possibly non-monotonic access structure $NM(\mathbb{A})$ over the set $\tilde{\mathcal{P}}$ of all unprimed parties in $\mathcal{P}$. For every set $\tilde{S} \subset \tilde{\mathcal{P}}$, $N(\tilde{S})$ is defined as $N(\tilde{S}) = \tilde{S} \cup \{x' | x \in \tilde{\mathcal{P}} \backslash \tilde{S}\}$. Then, $NM(\mathbb{A})$ is defined by saying that $\tilde{S}$ is authorized in $NM(\mathbb{A})$ if and only if $N(\tilde{S})$ is authorized in $\mathbb{A}$. For each access set $X \in NM(\mathbb{A})$, there is a set in $\mathbb{A}$ containing the elements in X and primed elements for each party not in X.

Key-(Ciphertext) Policy Attribute-Based Encrypion. Let $\mathcal{U} = \{0,1\}^*$ be an attribute space and $N = (n, \varphi)$ specify the corresponding bounds (the maximum numbers) on the size of attribute sets, the number of allowed repetition of same attributes which appear in a policy, respectively. Let $\mathcal{AS}_\varphi$ be a collection of access structures over $\mathcal{U}$ such that every access structure in $\mathcal{AS}_\varphi$ is specified by an access formula in which same attributes do not appear more than φ times. A bounded key (resp. ciphertext)-policy attribute-based encryption for $\mathcal{AS}_\varphi$ is a predicate encryption for $R^{\mathsf{KP}}_{(n,\varphi)} : \mathcal{AS}_\varphi \times \binom{\mathcal{U}}{<n} \to \{0,1\}$ (resp. $R^{\mathsf{CP}}_{(n,\varphi)} : \binom{\mathcal{U}}{<n} \times \mathcal{AS}_\varphi \to \{0,1\}$) defined by $R^{\mathsf{KP}}_{(n,\varphi)}(\mathbb{A}, \omega) = 1$ (resp. $R^{\mathsf{CP}}_{(n,\varphi)}(\omega, \mathbb{A}) = 1$) iff $\omega \in \mathbb{A}$ (for $\omega \subseteq U$ such that $|\omega| < n$ and $\mathbb{A} \in \mathcal{AS}_\varphi$). Let $\mathcal{AS}$ be a collection of access structure over $\mathcal{U}$. An unbounded key (resp., ciphertext)-policy attribute-based encryption scheme is a predicate encryption for $R^{\mathsf{KP}} : \mathcal{AS} \times 2^{\mathcal{U}} \to \{0,1\}$ (resp., $R^{\mathsf{CP}} : 2^{\mathcal{U}} \times \mathcal{AS} \to \{0,1\}$) defined by $R^{\mathsf{KP}}(\mathbb{A}, \omega) = 1$ (resp. $R^{\mathsf{CP}}(\omega, \mathbb{A}) = 1$) iff $\omega \in \mathbb{A}$ (for $\omega \subseteq U$ and $\mathbb{A} \in \mathcal{AS}$).

We note that the scheme of [27] (which was called unbounded ABE) can achieve the unbounded attribute set size, but it is still limited to the number of allowed repetition. Currently, only few KP-ABE schemes that are unbounded in full sence are known [21, 23, 30]. Note that the scheme in [21] uses random oracle model. In the CP-ABE setting, only scheme that is unbounded in full sense is recently proposed [30].

2.4 Number Theoretic Assumptions

We use groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order p with an efficiently computable mapping $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ s.t. $e(g^a, h^b) = e(g, h)^{ab}$ for any $(g, h) \in \mathbb{G} \times \mathbb{G}$, $a, b \in \mathbb{Z}$ and $e(g, h) \neq 1_{\mathbb{G}_T}$ whenever $g, h \neq 1_{\mathbb{G}}$.

Decisional Bilinear Diffie-Hellman (DBDH) Assumption. We say that an adversary $\mathcal{A}$ breaks the DBDH assumption on $(\mathbb{G}, \mathbb{G}_T)$ if $\mathcal{A}$ runs in polynomial time and $\frac{1}{2}|\Pr[\mathcal{A}(g, g^a, g^b, g^s, e(g,g)^{abs}) \to 0] - \Pr[\mathcal{A}(g, g^a, g^b, g^s, T) \to 0]|$ is negligible where $g \xleftarrow{\$} \mathbb{G}$, $T \xleftarrow{\$} \mathbb{G}_T$, $a, b, s \xleftarrow{\$} \mathbb{Z}_p$.

n-Decisional Bilinear Diffie-Hellman Exponent (n-DBDHE) Assumption [7]. We say that an adversary $\mathcal{A}$ breaks the n-DBDHE assumption on $(\mathbb{G}, \mathbb{G}_T)$ if $\mathcal{A}$ runs in polynomial time and

$\frac{1}{2}|\Pr[\mathcal{A}(g, \{g^{a^i}\}_{i\in[2n]\setminus\{n+1\}}, g^s, e(g,g)^{s\cdot a^{n+1}}) \to 0] - \frac{1}{2}|\Pr[\mathcal{A}(g, \{g^{a^i}\}_{i\in[2n]\setminus\{n+1\}}, g^s, T) \to 0]|$ is negligible where $g \xleftarrow{\$} \mathbb{G}$, $T \xleftarrow{\$} \mathbb{G}_T$, $a, s \xleftarrow{\$} \mathbb{Z}_p$.

3 Linear Two-Mode Identity Based Broadcast Encryption and Conversion to Non-monotonic KP-ABE

In this section, we first introduce the two mode inner product encryption scheme (TIPE) and two mode identity based broadcast encryption schemes (TIBBE) and explain how the latter can be derived from the former. Then, we propose a general transformation that transforms any TIBBE scheme that satisfies a certain condition into a non-monotonic KP-ABE scheme. Our transformation is an extension of the generic transformation proposed in [3], which converts any IBBE scheme with certain conditions into (monotonic) KP-ABE scheme.

3.1 Definition of TIPE and TIBBE

In a TIPE scheme, a ciphertext is associated with a vector $\mathbf{y}$. A private key is associated with $\mathsf{type} \in \{\mathsf{ZIPE}, \mathsf{NIPE}\}$ and a vector $\mathbf{x}$. Decryption is possible iff $\mathsf{type} = \mathsf{ZIPE}$ and $\langle \mathbf{x}, \mathbf{y} \rangle = 0$, or $\mathsf{type} = \mathsf{NIPE}$ and $\langle \mathbf{x}, \mathbf{y} \rangle \neq 0$. In a TIBBE scheme, a ciphertext is associated with a set of identities S. A private key is associated with $\mathsf{type} \in \{\mathsf{IBBE}, \mathsf{IBR}\}$ and an identity ID. Decryption is possible iff $\mathsf{type} = \mathsf{IBBE}$ and $\mathsf{ID} \in S$, or $\mathsf{type} = \mathsf{IBR}$ and $\mathsf{ID} \notin S$.

Here, we formally define TIPE and TIBBE as instances of PE as follows.

Two-Mode Inner Product Encryption scheme. TIPE is a predicate encryption for $R^{\mathsf{TIPE}}_{(n,p)} : (\mathbb{Z}_p^n \times \{\mathsf{ZIPE}, \mathsf{NIPE}\}) \times \mathbb{Z}_p^n \to \{0,1\}$ defined by $R^{\mathsf{TIPE}}_{(n,p)}((\mathbf{x}, \mathsf{type}), \mathbf{y}) = 1$ iff $(\langle \mathbf{x}, \mathbf{y} \rangle = 0 \wedge \mathsf{type} = \mathsf{ZIPE}) \vee (\langle \mathbf{x}, \mathbf{y} \rangle \neq 0 \wedge \mathsf{type} = \mathsf{NIPE})$.

Two-Mode Identity Based Broadcast Encryption scheme. TIBBE is a predicate encryption for $R^{\mathsf{TIBBE}}_n : (\mathcal{I} \times \{\mathsf{IBBE}, \mathsf{IBR}\}) \times \binom{\mathcal{I}}{<n} \to \{0,1\}$ defined by $R^{\mathsf{TIBBE}}_n((\mathsf{ID}, \mathsf{type}), S) = 1$ iff $(\mathsf{ID} \in S \wedge \mathsf{type} = \mathsf{IBBE}) \vee (\mathsf{ID} \notin S \wedge \mathsf{type} = \mathsf{IBR})$.

In later sections, we construct TIPE schemes instead of TIBBE schemes when it is simpler to describe. TIBBE scheme can be derived from TIPE scheme by the following technique due to [18]. The setup algorithm of the TIBBE scheme is the same as TIPE scheme. To generate a private key for $(\mathsf{ID}, \mathsf{IBBE})$ (resp. $(\mathsf{ID}, \mathsf{IBR})$), one runs key generation algorithm of TIPE scheme to obtain a private key for $(\mathbf{x}, \mathsf{ZIPE})$ (resp. $(\mathbf{x}, \mathsf{NIPE})$) where $\mathbf{x} = (1, \mathsf{ID}, \ldots, \mathsf{ID}^{n-1})$. To encrypt a message M for a set $S = (\mathsf{ID}_1, \ldots, \mathsf{ID}_k)$, one defines $\mathbf{y} = (y_1, \ldots, y_n)$ as a coefficient vector from

$$P_S[Z] = \sum_{i=1}^{k+1} y_i Z^{i-1} = \prod_{\mathsf{ID}_j \in S} (Z - \mathsf{ID}_j)$$

where, if $k+1 < n$, the coordinates $y_{k+1}, \ldots, y_n$ are all set to 0. Then, one runs encryption algorithm of TIPE scheme to encrypt M for a vector $\mathbf{y}$. To decrypt a ciphertext, one first defines $\mathbf{x}$ and $\mathbf{y}$ as above and runs the decryption algorithm of the TIPE scheme. Since $\mathsf{ID} \in S \Leftrightarrow P_S(\mathsf{ID}) = 0 \Leftrightarrow \langle \mathbf{x}, \mathbf{y} \rangle = 0$, the correctness of the resulting TIBBE scheme follows from the correctness of the underlying TIPE scheme. Furthermore, by the embedding lemma [8], the resulting TIBBE scheme is selectively secure if the underlying TIPE scheme is selectively secure.

3.2 Linear Two-Mode Identity Based Broadcast Encryption Template

We define a template for two-mode IBBE schemes that ensures that they give rise to selective secure non-monotonic KP-ABE schemes. We call this a linear TIBBE template. Let $\mathbb{G}, \mathbb{G}_T$ be

underlying bilinear groups of order p. The identity space of the scheme is $\mathcal{I} = \mathbb{Z}_p$. A linear TIBBE scheme is determined by parameters $n, n_1, n_2, \bar{n}_1, \in \mathbb{N}$, a distribution $\mathcal{G}$ on vectors of functions, and functions $\mathcal{D}^{\mathsf{IBBE}}$, $\mathcal{D}^{\mathsf{IBR}}$. $\mathcal{G}$'s output is tuple of functions $(f_1^{\mathsf{IBBE}}, f_2^{\mathsf{IBBE}}, f_1^{\mathsf{IBR}}, f_2^{\mathsf{IBR}}, F)$ where $f_1^{\mathsf{IBBE}} : \mathcal{I} \to \mathbb{G}, f_2^{\mathsf{IBBE}} : \mathcal{I} \to \mathbb{G}^{n_1}, f_1^{\mathsf{IBR}} : \mathcal{I} \to \mathbb{G}, f_2^{\mathsf{IBR}} : \mathcal{I} \to \mathbb{G}^{\bar{n}_1}, F : (\mathcal{I})^{\leq n-1} \times \mathbb{Z}_p \to \mathbb{G}^{\leq n_2}$. Here, we allow F to be probabilistic whereas all other functions are assumed to be deterministic. $\mathcal{D}^{\mathsf{IBBE}}$ and $\mathcal{D}^{\mathsf{IBR}}$ are functions such that $\mathcal{D}^{\mathsf{IBBE}} : \mathbb{G}^{n_1+1} \times \mathcal{I} \times \mathbb{G}^{n_2} \times \binom{\mathcal{I}}{<n} \to \mathbb{G}_T$, $\mathcal{D}^{\mathsf{IBR}} : \mathbb{G}^{\bar{n}_1+1} \times \mathcal{I} \times \mathbb{G}^{n_2} \times \binom{\mathcal{I}}{<n} \to \mathbb{G}_T$.

$\mathsf{Setup}(\lambda, n)$: Given a security parameter $\lambda \in \mathbb{N}$ and a bound $n \in \mathbb{Z}$ on the number of identities per ciphertext, the algorithm selects bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ and a generator $g \xleftarrow{\$} \mathbb{G}$. It computes $e(g,g)^\alpha$ for a random $\alpha \xleftarrow{\$} \mathbb{Z}_p$ and chooses functions $(f_1^{\mathsf{IBBE}}, f_2^{\mathsf{IBBE}}, f_1^{\mathsf{IBR}}, f_2^{\mathsf{IBR}}, F) \xleftarrow{\$} \mathcal{G}$. The master secret key consists of $\mathsf{msk} = \alpha$ while the master public key is $\mathsf{mpk} = (g, e(g,g)^\alpha, \{f_1^{\mathsf{type}}, f_2^{\mathsf{type}}\}_{\mathsf{type} \in \{\mathsf{IBBE}, \mathsf{IBR}\}}, F, n, n_1, n_2, \bar{n}_1)$.

$\mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, (\mathsf{ID}, \mathsf{type}))$: To generate a private key for ID of type $\mathsf{type} \in \{\mathsf{IBBE}, \mathsf{IBR}\}$, it chooses $r \xleftarrow{\$} \mathbb{Z}_p$. Then, it computes the private key as

$$\mathsf{sk}_{(\mathsf{ID},\mathsf{type})} = (d_1, d_2) = \Big(g^\alpha \cdot f_1^{\mathsf{type}}(\mathsf{ID})^r, f_2^{\mathsf{type}}(\mathsf{ID})^r\Big).$$

$\mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}, S)$: To encrypt $\mathsf{M} \in \mathbb{G}_T$ for a set of identities $S = (\mathsf{ID}_1, \ldots, \mathsf{ID}_k)$ where $k < n$, it chooses $s \xleftarrow{\$} \mathbb{Z}_p$ and computes the ciphertext as

$$C = (C_0, C_1) = (\mathsf{M} \cdot e(g,g)^{\alpha s}, F(\mathsf{ID}_1, \ldots, \mathsf{ID}_k, s)).$$

$\mathsf{Decrypt}(\mathsf{mpk}, C, S, \mathsf{sk}_{(\mathsf{ID},\mathsf{type})})$: It parses $\mathsf{sk}_{(\mathsf{ID},\mathsf{type})} = (d_1, d_2)$ and $C = (C_0, C_1)$ then runs

$$\mathcal{D}^{\mathsf{type}}\big((d_1, d_2), \mathsf{ID}, C_1, S\big) \to e(g,g)^{\alpha \cdot s},$$

and obtains $\mathsf{M} = C_0 / e(g,g)^\alpha$.

We also require that for all $(f_1^{\mathsf{IBBE}}, f_2^{\mathsf{IBBE}}, f_1^{\mathsf{IBR}}, f_2^{\mathsf{IBR}}, F) \xleftarrow{\$} \mathcal{G}$, the following property must hold. [‡]

Correctness. For all $\alpha, r, s \in \mathbb{Z}_p$, randomness for F, $(\mathsf{ID}, \mathsf{type}) \in \mathcal{I} \times \{\mathsf{IBBE}, \mathsf{IBR}\}$, $S = \{\mathsf{ID}_1, \ldots, \mathsf{ID}_k\} \in \binom{\mathcal{I}}{<n}$ such that $(\mathsf{type} = \mathsf{IBBE} \wedge \mathsf{ID} \in S) \vee (\mathsf{type} = \mathsf{IBR} \wedge \mathsf{ID} \notin S)$ and randomness for F, we have

$$\mathcal{D}^{\mathsf{type}}\Big(\big(g^\alpha \cdot f_1^{\mathsf{type}}(\mathsf{ID})^r, f_2^{\mathsf{type}}(\mathsf{ID})^r\big), \mathsf{ID}, F(\mathsf{ID}_1, \ldots, \mathsf{ID}_k, s), S\Big) = e(g,g)^{\alpha \cdot s}.$$

3.3 Generic Conversion from Linear TIBBE to Non-monotonic KP-ABE

Let $\Pi_{\mathsf{TIBBE}} = (\mathsf{Setup}', \mathsf{Keygen}', \mathsf{Encrypt}', \mathsf{Decrypt}')$ be a linear TIBBE system. We construct a non-monotonic KP-ABE scheme from Π_{TIBBE} as follows.

$\mathsf{Setup}(\lambda, n)$: It simply outputs $\mathsf{Setup}'(\lambda, n) \to (\mathsf{mpk}, \mathsf{msk})$.

$\mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, \tilde{\mathbb{A}})$: The input to the algorithm is the master secret key msk, the master public key mpk, and a non-monotonic access structure $\tilde{\mathbb{A}}$ such that we have $\tilde{\mathbb{A}} = NM(\mathbb{A})$ for some monotonic access structure $\mathbb{A}$ over a set $\mathcal{P}$ of attributes and associated with a linear secret sharing scheme (L, π). Let L be an $\ell \times m$ matrix. First, it generates shares of α with (L, π). Namely, it chooses a vector $\mathbf{s} = (s_1, \ldots, s_m)$ such that $s_1 = \alpha$ and $s_2, \ldots, s_m \xleftarrow{\$} \mathbb{Z}_p$ and calculates $\lambda_i = \langle \mathbf{L}_i, \mathbf{s} \rangle$ for each $i = 1, \ldots, \ell$. The party corresponds to share λ_i is $\pi(i) = \breve{x}_i$,

[‡]In [3], the authors also assume a property called linearity. However, we do not need this property.

where x_i is underlying attribute, and can be primed (i.e., negated) or unprimed (non-negated). Then for each $i = 1, \ldots, \ell$, it picks $r_i \xleftarrow{\$} \mathbb{Z}_p$ and sets D_i for each $i = 1, \ldots, \ell$ as follows.

$$D_i = \begin{cases} \left(d'_{i,1} = g^{\lambda_i} \cdot f_1^{\mathsf{IBBE}}(x_i)^{r_i},\ d'_{i,2} = f_2^{\mathsf{IBBE}}(x_i)^{r_i}\right) & \text{if } \pi(i) = x_i \\ \left(d'_{i,1} = g^{\lambda_i} \cdot f_1^{\mathsf{IBR}}(x_i)^{r_i},\ d'_{i,2} = f_2^{\mathsf{IBR}}(x_i)^{r_i}\right) & \text{if } \pi(i) = x'_i. \end{cases}$$

It then outputs the private key as $\mathsf{sk}_{\tilde{\mathbb{A}}} = \{D_i\}_{i=1}^{\ell}$

$\mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}, \omega)$: It simply outputs $\mathsf{Encrypt}'(\mathsf{mpk}, \mathsf{M}, \omega)$.

$\mathsf{Decrypt}(\mathsf{mpk}, C, \omega, \mathsf{sk}_{\tilde{\mathbb{A}}})$: Assume first that the policy $\tilde{\mathbb{A}}$ is satisfied by the attribute set ω, so that decryption is possible. Since $\tilde{\mathbb{A}} = NM(\mathbb{A})$ for some access structure $\mathbb{A}$ associated with a linear secret sharing scheme (L, π), we have $\omega' = N(\omega) \in \mathbb{A}$ and we let $I = \{i | \pi(i) \in \omega'\}$. Since ω' is authorized in $\mathbb{A}$, the receiver can efficiently compute reconstruction coefficients $\{(i, \mu_i)\}_{i \in I} = \mathsf{Recon}_{L,\pi}(\omega')$ such that $\sum_{i \in I} \mu_i \lambda_i = \alpha$. It parses $C = (C_0, C_1)$, $\mathsf{sk}_{\tilde{\mathbb{A}}} = \{D_i\}_{i=1}^{\ell}$ where $D_i = (d'_{i,1}, d'_{i,2})$ and computes $e(g,g)^{s \cdot \lambda_i}$ for each $i \in I$ as follows. (The correctness is shown later.)

$$\begin{cases} \mathcal{D}^{\mathsf{IBBE}}\big((d'_{i,1}, d'_{i,2}), x_i, C_1, \omega\big) \to e(g,g)^{s \cdot \lambda_i} & \text{if } \pi(i) = x_i \quad (1a) \\ \mathcal{D}^{\mathsf{IBR}}\big((d'_{i,1}, d'_{i,2}), x_i, C_1, \omega\big) \to e(g,g)^{s \cdot \lambda_i} & \text{if } \pi(i) = x'_i. \quad (1b) \end{cases}$$

Finally, it recovers message by $C_0 \cdot \prod_{i \in I} \big(e(g,g)^{s \cdot \lambda_i}\big)^{-\mu_i} = \mathsf{M}$.

Correctness. We now verify that equations (1a) and (1b) are correct. (1a) and (1b) follow from the correctness of the underlying TIBBE scheme by seeing D_i as a private key for $\big(\mathsf{ID} = x_i, \mathsf{type} \in \{\mathsf{IBBE}, \mathsf{IBR}\}\big)$ that is derived from $\mathsf{msk} = \lambda_i$ using randomness r_i.

The security of the resulting scheme is established by the following Theorem. We prove it in Appendix B.

Theorem 1. *If the underlying TIBBE scheme is selectively secure, then the resulting KP-ABE system above is also selectively secure.*

Remark. We have described the conversion for TIBBE scheme with a restriction that the number of identities per ciphertext is bounded by n. However, the same conversion also applies to a TIBBE scheme without such a restriction. In particular, we can apply the above conversion to our TIBBE scheme in Sec. 6.

4 TIPE Scheme with Compact Ciphertexts

In this section, we propose a TIPE scheme with compact ciphertext size. As we explained in Sec. 3.1, we can obtain a TIBBE scheme from the TIPE scheme. By applying the conversion in Sec. 3 to this TIBBE scheme, we obtain a new non-monotonic KP-ABE scheme with very short ciphertexts. We give a concrete description of the resulting scheme in Appendix C. The ciphertext overhead is 33% shorter than the non-monotonic KP-ABE ciphertext in [3] (the shortest in the literature). It also reduces the number of pairing operations in the decryption algorithm from 3 to 2. The public key size of our scheme is about half that of the existing scheme, but the private key of our scheme is slightly longer.

$\mathsf{Setup}(\lambda, n)$: It chooses bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ with $g \xleftarrow{\$} \mathbb{G}$. It also picks $v, \alpha \xleftarrow{\$} \mathbb{Z}_p$ and $\mathbf{u} = (u_1, \ldots, u_n) \xleftarrow{\$} \mathbb{Z}_p^n$. Then it sets $V = g^v$ and $U = (U_1, \ldots, U_n) = g^{\mathbf{u}}$. It finally outputs the master public key $\mathsf{mpk} = (g, U_1, \ldots, U_n, V, e(g,g)^\alpha)$ and the master secret key $\mathsf{msk} = \alpha$.

$\mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, (\mathbf{x}, \mathsf{type}))$: To generate a private key for $\big(\mathbf{x} = (x_1 \neq 0, \ldots, x_n) \in \mathbb{Z}_p^* \times \mathbb{Z}_p^{n-1}, \mathsf{type} \in \{\mathsf{ZIPE}, \mathsf{NIPE}\}\big)$, it chooses $r \xleftarrow{\$} \mathbb{Z}_p$ and computes

$$\begin{cases} \mathsf{sk}_{(\mathbf{x},\mathsf{ZIPE})} = \Big(D_1 = g^\alpha V^r, D_2 = g^r, \{K_i = (U_1^{-\frac{x_i}{x_1}} U_i)^r\}_{i=2,\ldots,n}\Big) & \text{if } \mathsf{type} = \mathsf{ZIPE} \\ \mathsf{sk}_{(\mathbf{x},\mathsf{NIPE})} = \Big(D_1 = g^\alpha U_1^r, D_2 = g^r, D_3 = V^r, \{K_i = (U_1^{-\frac{x_i}{x_1}} U_i)^r\}_{i=2,\ldots,n}\Big) & \text{if } \mathsf{type} = \mathsf{NIPE}. \end{cases}$$

$\mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}, \mathbf{y})$: To encrypt $\mathsf{M} \in \mathbb{G}_T$ for the vector $\mathbf{y} = (y_1, \ldots, y_n) \in \mathbb{Z}_p^n$, it picks $s \xleftarrow{\$} \mathbb{Z}_p$ and computes the ciphertext as

$$C = \Big(C_0 = \mathsf{M} \cdot e(g,g)^{\alpha s}, C_1 = g^s, C_2 = (V U_1^{y_1} \cdots U_n^{y_n})^{-s}\Big).$$

$\mathsf{Decrypt}(\mathsf{mpk}, C, \mathbf{y}, \mathsf{sk}_{(\mathbf{x},\mathsf{type})})$**:** It computes

$$\begin{cases} e(C_1, D_1 \cdot \prod_{i=2}^n K_i^{y_i}) \cdot e(C_2, D_2) = e(g,g)^{s\alpha} & \text{if } \mathsf{type} = \mathsf{ZIPE} \\ e(C_1, D_1) \cdot \Big(e(C_1, D_3 \prod_{i=2}^n K_i^{y_i}) \cdot e(C_2, D_2)\Big)^{\frac{x_1}{\langle \mathbf{x}, \mathbf{y} \rangle}} = e(g,g)^{s\alpha} & \text{if } \mathsf{type} = \mathsf{NIPE} \end{cases}$$

and recovers the message by $C_0 / e(g,g)^{s\alpha} = \mathsf{M}$.

The correctness of the scheme is proven in Appendix D.

We construct the above scheme by combining the IPE scheme derived from the spatial encryption scheme in [8, 2] and a variant of the NIPE scheme proposed in [3] so that they share the master public key and the ciphertext. The non-monotonic KP-ABE scheme derived from the above TIPE scheme has compact parameters, because of this share of parameters. The main technical challenge in the proof of the security of the scheme is to simulate the key generation oracle for two different types (i.e., ZIPE and NIPE) of keys *simultaneously*. To achieve this, we use a significantly different strategy to simulate NIPE keys than the security proof in [3]. The following theorem addresses the security of the scheme.

Theorem 2. *The above TIPE scheme is selectively secure under the n-DBDHE assumption.*

Before proving the theorem, we recall following useful lemma that is implicit in [8]. We prove the lemma in Appendix A.

Lemma 1. *([8]) Let $\mathbb{G}$ be a multiplicative group with prime order p and g be its generator. Let n, m be some integer bounded by polynomial of λ, $\mathbf{a}$ be $\mathbf{a} = (a, a^2, \ldots, a^n) \in \mathbb{Z}_p^n$, $\tilde{\alpha}, \{w_i\}_{i=0}^m$ be elements in $\mathbb{Z}_p$, $\{\mathbf{z}_i\}_{i=0}^m$ be vectors in $\mathbb{Z}_p^n$. We also assume that $\mathbf{h} = (h_1, \ldots, h_n) \in \mathbb{Z}_p^n$ satisfies $\langle \mathbf{h}, \mathbf{z}_0 \rangle \neq 0$ and $\langle \mathbf{h}, \mathbf{z}_i \rangle = 0$ for $i \in [m]$. Then, there exists an PPT* BHSim *which takes $(\tilde{\alpha}, \{\mathbf{z}_i\}_{i=0}^m, \{w_i\}_{i=0}^m, \mathbf{h}, \{g^{a^i}\}_{i \in [2n] \setminus \{n+1\}})$ as input and outputs $(g^{a^{n+1}+\tilde{\alpha}} \cdot (g^{\langle \mathbf{z}_0, \mathbf{a} \rangle + w_0})^r, \{(g^{\langle \mathbf{z}_i, \mathbf{a} \rangle + w_i})^r\}_{i=1}^m)$ where $r \xleftarrow{\$} \mathbb{Z}_p$.*

Proof. (of Theorem 2.) We construct an algorithm $\mathcal{B}$ that receives $(g, \{g^{a^i}\}_{i\in[2n]\setminus\{n+1\}}, g^s, T) \in \mathbb{G}^{2n+1} \times \mathbb{G}_T$ and decides if $T = e(g,g)^{a^{n+1}s}$ using the selective adversary $\mathcal{A}$ against our scheme. We denote by $\mathbf{a}$ a vector $(a, a^2, \ldots, a^n)$.

Setup of master public key. At the outset of the game, the adversary $\mathcal{A}$ declares the challenge vector $\mathbf{y}^\star = (y_1^\star, \ldots, y_n^\star) \in \mathbb{Z}_p^n$. $\mathcal{B}$ picks $\tilde{\alpha}, \tilde{v} \xleftarrow{\$} \mathbb{Z}_p$, $\tilde{\mathbf{u}} = (\tilde{u}_1, \ldots, \tilde{u}_n) \xleftarrow{\$} \mathbb{Z}_p^n$ and sets mpk as

$$\mathsf{mpk} = \big(g = g, e(g,g)^\alpha = e(g^a, g^{a^n}) \cdot e(g,g)^{\tilde{\alpha}}, \mathbf{U} = g^{\mathbf{a}} \cdot g^{\tilde{\mathbf{u}}}, V = g^{-\langle \mathbf{a}, \mathbf{y}^\star\rangle} \cdot g^{\tilde{v}}\big),$$

and gives it to $\mathcal{A}$. Here, we implicitly set $\alpha = \tilde{\alpha} + a^{n+1}$, $\mathbf{u} = \mathbf{a} + \tilde{\mathbf{u}}$, and $v = -\langle \mathbf{a}, \mathbf{y}^\star\rangle + \tilde{v}$.

Phase1 and 2. When $\mathcal{A}$ queries private key for $(\mathbf{x} = (x_1, \ldots, x_n), \mathsf{type}) \in \mathbb{Z}_p^* \times \mathbb{Z}_p^{n-1} \times \{\mathsf{ZIPE}, \mathsf{NIPE}\}$, $\mathcal{B}$ answers as follows.

- If $\mathsf{type} = \mathsf{ZIPE}$, we have $\langle \mathbf{x}, \mathbf{y}^\star\rangle \neq 0$. In this case, $\mathcal{B}$ first sets $\mathbf{z}_0 = -\mathbf{y}^\star, \mathbf{z}_1 = \mathbf{0}, \mathbf{z}_i = -\frac{x_i}{x_1}\mathbf{e}_1 + \mathbf{e}_i$ for $i = 2, \ldots, n$, $w_0 = \tilde{v}, w_1 = 1$, and $w_i = -\frac{x_i}{x_1}\tilde{u}_1 + \tilde{u}_i$ for $i = 2, \ldots, n$. Then $\mathcal{B}$ runs $\mathsf{BHSim}(\tilde{\alpha}, \{\mathbf{z}_i\}_{i=0}^n, \{w_i\}_{i=0}^n, \mathbf{x}, \{g^{a^i}\}_{i\in[2n]\setminus\{n+1\}}) \to (Z_0, \{Z_i\}_{i=1}^n)$ and returns $(D_1, D_2, \{K_i\}_{i=2}^n) = (Z_0, Z_1, \{Z_i\}_{i=2}^n)$. We claim that $(D_1, D_2, \{K_i\}_{i=2}^n)$ is distributed the same as real private key. At first, we check that the input to BHSim is in a valid form. To see this, it suffices to check that $\langle \mathbf{x}, \mathbf{z}_0\rangle = \langle \mathbf{x}, -\mathbf{y}^\star\rangle \neq 0$, $\langle \mathbf{x}, \mathbf{z}_1\rangle = \langle \mathbf{x}, \mathbf{0}\rangle = 0$, and $\langle \mathbf{x}, \mathbf{z}_i\rangle = \langle \mathbf{x}, -\frac{x_i}{x_1}\mathbf{e}_1 + \mathbf{e}_i\rangle = -x_1 \cdot \frac{x_i}{x_1} + x_i = 0$ for $i = 2, \ldots, n$. Since the input to BHSim is in a valid form, $D_1 = Z_0 = g^{\tilde{\alpha}+a^{n+1}}(g^{-\langle \mathbf{a}, \mathbf{y}^\star\rangle} \cdot g^{\tilde{v}})^r = g^\alpha V^r$, $D_2 = Z_1 = (g^{\langle \mathbf{0}, \mathbf{a}\rangle + 1})^r = g^r$, and

$$K_i = Z_i = (g^{\langle -\frac{x_i}{x_1}\mathbf{e}_1 + \mathbf{e}_i, \mathbf{a}\rangle - \frac{x_i}{x_1}\tilde{u}_1 + \tilde{u}_i})^r = (g^{-\frac{x_i}{x_1}(a+\tilde{u}_1)} \cdot g^{a^i + \tilde{u}_i})^r = (U_1^{-\frac{x_i}{x_1}} \cdot U_i)^r$$

for $i \in \{2, \ldots, n\}$ where $r \xleftarrow{\$} \mathbb{Z}_p$ as desired.

- If $\mathsf{type} = \mathsf{NIPE}$, we have $\langle \mathbf{x}, \mathbf{y}^\star\rangle = 0$. In this case, $\mathcal{B}$ first sets $\mathbf{z}_0 = \mathbf{e}_1, \mathbf{z}_1 = \mathbf{0}, \mathbf{z}_i = -\frac{x_i}{x_1}\mathbf{e}_1 + \mathbf{e}_i$ for $i = 2, \ldots, n$, $\mathbf{z}_{n+1} = -\mathbf{y}^\star$, $w_0 = \tilde{u}_1, w_1 = 1$, $w_i = -\frac{x_i}{x_1}\tilde{u}_1 + \tilde{u}_i$ for $i = 2, \ldots, n$, and $w_{n+1} = \tilde{v}$. Then $\mathcal{B}$ runs $\mathsf{BHSim}(\tilde{\alpha}, \{\mathbf{z}_i\}_{i=0}^{n+1}, \{w_i\}_{i=0}^{n+1}, \mathbf{x}, \{g^{a^i}\}_{i\in[2n]\setminus\{n+1\}}) \to (Z_0, \{Z_i\}_{i=1}^{n+1})$ and returns $(D_1, D_2, D_3, \{K_i\}_{i=2}^n) = (Z_0, Z_1, Z_{n+1}, \{Z_i\}_{i=2}^n)$. We claim that $(D_1, D_2, D_3, \{K_i\}_{i=2}^n)$ is distributed the same as real private key. At first, we check that the input to BHSim is in a valid form. To see this, it suffices to check that $\langle \mathbf{x}, \mathbf{z}_0\rangle = \langle \mathbf{x}, \mathbf{e}_1\rangle = x_1 \neq 0$, $\langle \mathbf{x}, \mathbf{z}_1\rangle = \langle \mathbf{x}, \mathbf{0}\rangle = 0$, $\langle \mathbf{x}, \mathbf{z}_i\rangle = \langle \mathbf{x}, -\frac{x_i}{x_1}\mathbf{e}_1 + \mathbf{e}_i\rangle = 0$ for $i = 2, \ldots, n$, and $\langle \mathbf{x}, \mathbf{z}_{n+1}\rangle = \langle \mathbf{x}, -\mathbf{y}^\star\rangle = 0$. Since the input to BHSim is in a valid form, we have

$$D_1 = Z_0 = g^{\tilde{\alpha}+a^{n+1}} \cdot (g^{\langle \mathbf{a}, \mathbf{e}_1\rangle} \cdot g^{\tilde{u}_1})^r = g^\alpha \cdot (g^{a+\tilde{u}_1})^r = g^\alpha U_1^r$$

where $r \xleftarrow{\$} \mathbb{Z}_p$. We can also check that $D_2 = g^r$ and $\{K_i\}_{i=2}^n = \{(U_1^{-\frac{x_i}{x_1}} \cdot U_i)^r\}_{i=2}^n$ by exactly the same computation as in the case of $\mathsf{type} = \mathsf{ZIPE}$. Finally, we have that $D_3 = Z_{n+1} = (g^{-\langle \mathbf{a}, \mathbf{y}^\star\rangle + \tilde{v}})^r = V^r$ as desired.

Challenge. At some point in the game, $\mathcal{A}$ submits a pair of ciphertexts $(\mathsf{M}_0, \mathsf{M}_1)$ to $\mathcal{B}$. $\mathcal{B}$ flips a random coin $\beta \xleftarrow{\$} \{0,1\}$ and returns $(C_0, C_1, C_2) = (\mathsf{M}_\beta \cdot e(g^s, g^{\tilde{\alpha}}) \cdot T, g^s, (g^s)^{-(\langle \mathbf{y}^\star, \tilde{\mathbf{u}}\rangle + \tilde{v})})$ to $\mathcal{A}$. Since

$$(g^s)^{-(\langle \mathbf{y}^\star, \tilde{\mathbf{u}}\rangle + \tilde{v})} = (g^{\langle -\mathbf{a}, \mathbf{y}^\star\rangle + \tilde{v}} \cdot g^{\langle \mathbf{a}+\tilde{\mathbf{u}}, \mathbf{y}^\star\rangle})^{-s} = (V U_1^{y_1^\star} \cdots U_n^{y_n^\star})^{-s}$$

and $e(g^s, g^{\tilde{\alpha}}) \cdot e(g,g)^{a^{n+1}s} = e(g,g)^{s\alpha}$, the ciphertext is in a valid form if $T = e(g,g)^{a^{n+1}s}$.

Guess. Finally, $\mathcal{A}$ outputs its guess β' for β. If $\beta' = \beta$, $\mathcal{A}$ outputs 1 for its guess. Otherwise, it outputs 0. If $T = e(g,g)^{sa^{n+1}}$, the above simulation is perfect and thus $\mathcal{A}$ has non-negligible advantage. On the other hand, If T is a random element in $\mathbb{G}_T$, $\mathcal{A}$'s advantage is 0. Therefore, if $\mathcal{A}$ breaks our scheme with non-negligible advantage, $\mathcal{B}$ has a non-negligible advantage against the n-DBDHE assumption. □

5 TIPE Scheme from the DBDH assumption

In this section, we propose a TIPE scheme from the DBDH assumption, which is one of the weakest assumptions in bilinear groups. By sequentially applying the conversions from TIPE to TIBBE in Sec. 3.1 and from TIBBE to non-monotonic KP-ABE in Sec. 3 to the scheme, we obtain a new non-monotonic KP-ABE scheme from the DBDH assumption. We give a concrete description of the resulting scheme in Appendix C. Compared to the Non-monotonic KP-ABE scheme from the same assumption in [28], the public key and ciphertext size of our scheme are approximately half the size of theirs, and the private key size is comparable.

$\mathsf{Setup}(\lambda, n)$: It chooses bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ with $g \xleftarrow{\$} \mathbb{G}$. It also picks $u, \alpha \xleftarrow{\$} \mathbb{Z}_p$ and $\mathbf{v} = (v_1, \ldots, v_n) \xleftarrow{\$} \mathbb{Z}_p^n$. Then it sets $U = g^u$ and $\mathbf{V} = (V_1, \ldots, V_n) = g^{\mathbf{v}}$. It finally outputs the master public key $\mathsf{mpk} = (g, U, V_1, \ldots, V_n, e(g,g)^\alpha)$ and the master secret key $\mathsf{msk} = \alpha$.

$\mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}, \mathbf{y})$: To encrypt $\mathsf{M} \in \mathbb{G}_T$ for the vector $\mathbf{y} = (y_1, \ldots, y_n) \in \mathbb{Z}_p^n$, it picks $s \xleftarrow{\$} \mathbb{Z}_p$ and computes the ciphertext as

$$C = \Big(C_0 = \mathsf{M} \cdot e(g,g)^{\alpha s}, C_1 = g^s, \{E_i = (U^{y_i} V_i)^{-s}\}_{i=1,\ldots n}\Big).$$

$\mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, (\mathbf{x}, \mathsf{type}))$: To generate a private key for $\big(\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{Z}_p^n, \mathsf{type} \in \{\mathsf{ZIPE}, \mathsf{NIPE}\}\big)$, it chooses $r \xleftarrow{\$} \mathbb{Z}_p$ and computes

$$\begin{cases} \mathsf{sk}_{(\mathbf{x},\mathsf{ZIPE})} = \Big(D_1 = g^\alpha \cdot (V_1^{x_1} \cdots V_n^{x_n})^r, D_2 = g^r\Big) & \text{if } \mathsf{type} = \mathsf{ZIPE} \\ \mathsf{sk}_{(\mathbf{x},\mathsf{NIPE})} = \Big(D_1 = g^\alpha U^r, D_2 = (V_1^{x_1} \cdots V_n^{x_n})^r, D_3 = g^r\Big) & \text{if } \mathsf{type} = \mathsf{NIPE}. \end{cases}$$

$\mathsf{Decrypt}(\mathsf{mpk}, C, \mathbf{y}, \mathsf{sk}_{(\mathbf{x},\mathsf{type})})$**:** It computes

$$\begin{cases} e(C_1, D_1) \cdot e(\prod_{i=1}^n E_i^{x_i}, D_2) = e(g,g)^{s\alpha} & \text{if } \mathsf{type} = \mathsf{ZIPE} \\ e(C_1, D_1) \cdot \Big(e(\prod_{i=1}^n E_i^{x_i}, D_3) \cdot e(C_1, D_2)\Big)^{\frac{1}{\langle \mathbf{x},\mathbf{y}\rangle}} = e(g,g)^{s\alpha} & \text{if } \mathsf{type} = \mathsf{NIPE} \end{cases}$$

and recovers the message by $C_0 / e(g,g)^{s\alpha} = \mathsf{M}$.

The correctness of the scheme is proven in Appendix D. The following theorem addresses the security of the scheme.

Theorem 3. *The above TIPE scheme is selectively secure under the DBDH assumption.*

Before proving the theorem, we recall following useful lemma that is implicit in [6]. We prove the lemma in Appendix A.

Lemma 2. *([6]) Let $\mathbb{G}$ be a multiplicative group with prime order p and g be its generator. There exists an PPT BBSim which takes as $(A, B, z \neq 0, w) \in \mathbb{G}^2 \times \mathbb{Z}_p^* \times \mathbb{Z}_p$ where $(A, B) = (g^a, g^b)$ and outputs $(g^{ab} \cdot (A^z g^w)^r, g^r)$ where $r \xleftarrow{\$} \mathbb{Z}_p$.*

Proof. (of Theorem 3.) We construct an algorithm $\mathcal{B}$ that receives $(g, g^a, g^b, g^s, T) \in \mathbb{G}^4 \times \mathbb{G}_T$ and decides if $T = e(g,g)^{abs}$ using the selective adversary $\mathcal{A}$ against our scheme.

Setup of master public key. At the outset of the game, the adversary $\mathcal{A}$ declares challenge vector $\mathbf{y}^\star = (y_1^\star, \ldots, y_n^\star) \in \mathbb{Z}_p^n$. $\mathcal{B}$ picks $\tilde{\mathbf{v}} = (\tilde{v}_1, \ldots, \tilde{v}_n) \xleftarrow{\$} \mathbb{Z}_p^n$, sets mpk as

$$\mathsf{mpk} = \big(g = g, e(g,g)^\alpha = e(g^a, g^b), U = g^a, \mathbf{V} = (g^a)^{-\mathbf{y}^\star} \cdot g^{\tilde{\mathbf{v}}}\big),$$

and gives it to $\mathcal{A}$. Here, we implicitly set $\alpha = ab$, $u = a$, and $\mathbf{v} = -a\mathbf{y}^\star + \tilde{\mathbf{v}}$.

Phase1 and 2. When $\mathcal{A}$ queries private key for $(\mathbf{x} = (x_1, \ldots, x_n), \mathsf{type}) \in \mathbb{Z}_p^n \times \{\mathsf{ZIPE}, \mathsf{NIPE}\}$, $\mathcal{B}$ answers as follows.

- If $\mathsf{type} = \mathsf{ZIPE}$, $\mathcal{B}$ runs $\mathsf{BBSim}(g^a, g^b, -\langle \mathbf{x}, \mathbf{y}^\star \rangle, \langle \mathbf{x}, \tilde{\mathbf{v}} \rangle) \to (D_1, D_2)$ and returns (D_1, D_2) as a secret key for $(\mathbf{x}, \mathsf{ZIPE})$. Since $\mathsf{type} = \mathsf{ZIPE}$, $\langle \mathbf{x}, \mathbf{y}^\star \rangle \neq 0$ holds, the input to BBSim is in a valid form. The output of BBSim is $(D_1, D_2) = (g^{ab} \cdot ((g^a)^{-\langle \mathbf{x}, \mathbf{y}^\star \rangle} \cdot g^{\langle \mathbf{x}, \tilde{\mathbf{v}} \rangle})^r, g^r)$ for $r \xleftarrow{\$} \mathbb{Z}_p$. Since

$$D_1 = g^{ab} \cdot ((g^a)^{-\langle \mathbf{x}, \mathbf{y}^\star \rangle} \cdot g^{\langle \mathbf{x}, \tilde{\mathbf{v}} \rangle})^r = g^{ab} \cdot \Big(\prod_{i=1}^n ((g^a)^{-y_i^\star} \cdot g^{\tilde{v}_i})^{x_i}\Big)^r = g^\alpha (V_1^{x_1} \cdots V_n^{x_n})^r$$

  holds, (D_1, D_2) is distributed exactly the same as real private key.

- If $\mathsf{type} = \mathsf{NIPE}$, $\mathcal{B}$ first runs $\mathsf{BBSim}(g^a, g^b, 1, 0) \to (D_1, D_3)$. Then $\mathcal{B}$ computes $D_2 = D_3^{\langle \mathbf{x}, \tilde{\mathbf{v}} \rangle}$ and returns (D_1, D_2, D_3) as a secret key for $(\mathbf{x}, \mathsf{NIPE})$. Since the input to BBSim is in a valid form, the output of BBSim is $(D_1, D_3) = (g^{ab} \cdot (g^a)^r, g^r) = (g^\alpha \cdot U^r, g^r)$ for $r \xleftarrow{\$} \mathbb{Z}_p$. We can see that

$$D_2 = (g^r)^{\langle \mathbf{x}, \tilde{\mathbf{v}} \rangle} = \Big((g^a)^{-\langle \mathbf{x}, \mathbf{y}^\star \rangle} \cdot g^{\langle \mathbf{x}, \tilde{\mathbf{v}} \rangle}\Big)^r = \Big(\prod_{i=1}^n ((g^a)^{-y_i^\star} \cdot g^{\tilde{v}_i})^{x_i}\Big)^r = \big(V_1^{x_1} \cdots V_n^{x_n}\big)^r$$

  holds and thus (D_1, D_2, D_3) is distributed exactly the same as real private key. In the second equality above, we used the fact that $\langle \mathbf{x}, \mathbf{y}^\star \rangle = 0$.

Challenge. At some point in the game, $\mathcal{A}$ submits a pair of ciphertexts $(\mathsf{M}_0, \mathsf{M}_1)$ to $\mathcal{B}$. $\mathcal{B}$ flips a random coin $\beta \xleftarrow{\$} \{0,1\}$ and returns $(C_0, C_1, \{E_i\}_{i=1}^n) = (\mathsf{M}_\beta \cdot T, g^s, \{(g^s)^{-\tilde{v}_i}\}_{i=1}^n)$ to $\mathcal{A}$. Since

$$(g^s)^{-\tilde{v}_i} = \Big((g^a)^{y_i^\star} \cdot (g^a)^{-y_i^\star} \cdot g^{\tilde{v}_i}\Big)^{-s} = (U^{y_i^\star} \cdot V_i)^{-s}$$

for $i = 1, \ldots, n$, it can be seen that the ciphertext is in a valid form if $T = e(g,g)^{abs}$.

Finally, $\mathcal{A}$ outputs its guess β' for β. If $\beta' = \beta$, $\mathcal{A}$ outputs 1 for its guess. Otherwise, it outputs 0. If $T = e(g,g)^{abs}$, the above simulation is perfect and thus $\mathcal{A}$ has non-negligible advantage. On the other hand, if T is a random element in $\mathbb{G}_T$, $\mathcal{A}$'s advantage is 0. Therefore, if $\mathcal{A}$ breaks our scheme with non-negligible advantage, $\mathcal{B}$ has a non-negligible advantage against the DBDH assumption. □

6 Unbounded TIBBE Scheme

In the TIBBE schemes derived from the TIPE schemes in Sec. 4 and 5, the number of identities per ciphertext is bounded by a parameter n. In this section, we propose a TIBBE scheme without such a restriction. The structure of the construction can be seen as a combination of the IBBE scheme implicit in KP-ABE scheme in [30] and the IBR scheme in [21]. By applying the conversion in Sec. 3 to the scheme, we obtain the first non-monotonic KP-ABE scheme in the standard model that does not restrict the number of attributes per ciphertext or the number of times the same attribute can be used in an access formula associated with a private key. We give a concrete description of the resulting scheme in Appendix C.

$\mathsf{Setup}(\lambda)$: It chooses bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ with $g \xleftarrow{\$} \mathbb{G}$. It also picks $H, U, V, W \xleftarrow{\$} \mathbb{G}$ and $b, \alpha \xleftarrow{\$} \mathbb{Z}_p^n$. Then it sets $B = g^b, B' = g^{b^2}, V' = V^b$. It finally outputs the master public key $\mathsf{mpk} = (g, H, U, W, B, B', V, V', e(g,g)^\alpha)$ and the master secret key $\mathsf{msk} = \alpha$.

$\mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}, S)$: To encrypt $\mathsf{M} \in \mathbb{G}_T$ for the set of identities $S = (\mathsf{ID}_1, \ldots, \mathsf{ID}_k) \subset \mathbb{Z}_p$, it chooses $s, t_1, \ldots, t_k \xleftarrow{\$} \mathbb{Z}_p$ and random $s_1, \ldots, s_k \in \mathbb{Z}_p$ such that $s_1 + \ldots + s_k = s$ and computes the ciphertext as

$$C = \left(C_0 = \mathsf{M} \cdot e(g,g)^{\alpha s}, C_1 = g^s, \left\{ \begin{array}{ll} C_{i,1} = W^{-s}(U^{\mathsf{ID}_i} H)^{-t_i}, & C_{i,2} = g^{t_i} \\ C'_{i,1} = (B'^{\mathsf{ID}_i} V')^{-s_i}, & C'_{i,2} = B^{s_i} \end{array} \right\}_{i \in [k]} \right).$$

$\mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, (\mathsf{ID}, \mathsf{type}))$: To generate a private key for $\mathsf{ID} \in \mathbb{Z}_p$, it chooses $r \xleftarrow{\$} \mathbb{Z}_p$ and computes the private key as

$$\begin{cases} \mathsf{sk}_{(\mathsf{ID},\mathsf{IBBE})} = \left(D_1 = g^\alpha \cdot W^r, D_2 = (U^{\mathsf{ID}} H)^r, D_3 = g^r \right) & \text{if } \mathsf{type} = \mathsf{IBBE} \\ \mathsf{sk}_{(\mathsf{ID},\mathsf{IBR})} = \left(D_1 = g^\alpha \cdot (B')^r, D_2 = (B^{\mathsf{ID}} V)^r, D_3 = g^r \right) & \text{if } \mathsf{type} = \mathsf{IBR}. \end{cases}$$

$\mathsf{Decrypt}(\mathsf{mpk}, C, S, \mathsf{sk}_{(\mathsf{ID},\mathsf{type})})$**:** We assume that in the case of $\mathsf{type} = \mathsf{IBBE}$, ID is contained in $\mathsf{ID} \in S = \{\mathsf{ID}_1, \ldots, \mathsf{ID}_k\}$, so that decryption is possible. Therefore, there is an $\tau \in [k]$ such that $\mathsf{ID} = \mathsf{ID}_\tau$. It computes

$$\begin{cases} e(C_1, D_1) \cdot e(C_{\tau,1}, D_3) \cdot e(C_{\tau,2}, D_2) = e(g,g)^{s\alpha} & \text{if } \mathsf{type} = \mathsf{IBBE} \\ e(C_1, D_1) \cdot \prod_{i=1}^{k} \left(e(C'_{i,1}, D_3) \cdot e(C'_{i,2}, D_2) \right)^{1/(\mathsf{ID}_i - \mathsf{ID})} = e(g,g)^{s\alpha} & \text{if } \mathsf{type} = \mathsf{IBR} \end{cases}$$

and recovers the message by $C_0 / e(g,g)^{s\alpha} = \mathsf{M}$.

The correctness of the scheme is proven in Appendix D. We can prove selective security of the scheme under the new assumption that we call n-(A) assumption which is secure in the generic group model. The definition of the assumption and the proof appear in Appendix E.

7 Unbounded Non-monotonic CP-ABE Scheme

In this section, we propose the first non-monotonic CP-ABE scheme that does not restrict the size of the attributes set or the number of times the same attribute can be used in an access formula. Our starting point for the construction of the scheme is the unbounded (monotonic) CP-ABE scheme in [30]. To support the non-monotonic access structure, we first construct a suitable revocation mechanism, which can be seen as a ciphertext-policy version of the IBR scheme in [21]. Then, we combine this with the CP-ABE scheme in [30] to obtain our new scheme. Because some parameters are shared between the two schemes, the public key of our scheme is only one group element longer than that of the scheme in [30], while our scheme supports a more general access structure.

$\mathsf{Setup}(\lambda)$: It chooses bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ with $g \xleftarrow{\$} \mathbb{G}$. It also picks $b, \alpha \xleftarrow{\$} \mathbb{Z}_p$ and $H, U, V, W \xleftarrow{\$} \mathbb{G}$. Then it sets $V' = U^b$ and outputs the master public key $\mathsf{mpk} = (g, H, U, V, V', W, e(g,g)^\alpha)$ and the master secret key $\mathsf{msk} = (\alpha, b)$.

$\mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, \omega)$: To generate a private key for a set of attributes $\omega = \{\omega_1, \ldots \omega_k\} \subset \mathbb{Z}_p$, it chooses $r, r_1, \ldots, r_k \xleftarrow{\$} \mathbb{Z}_p$ and random $r'_1, \ldots, r'_k \in \mathbb{Z}_p$ such that $r'_1 + \ldots + r'_k = r$. It then outputs the private key as

$$\mathsf{sk}_\omega = \left(D_1 = g^\alpha W^r, D_2 = g^r, \left\{ \begin{array}{ll} K_{i,1} = V^{-r}(U^{\omega_i} H)^{r_i}, & K_{i,2} = g^{r_i} \\ K'_{i,1} = (U^{b\omega_i} H^b)^{r'_i}, & K'_{i,2} = g^{br'_i} \end{array} \right\}_{i \in [k]} \right).$$

$\mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}, \tilde{\mathbb{A}})$: The input to the algorithm is the master public key mpk, the message $\mathsf{M} \in \mathbb{G}_T$ and a non-monotonic access structure $\tilde{\mathbb{A}}$ such that we have $\tilde{\mathbb{A}} = NM(\mathbb{A})$ for some monotonic access structure $\mathbb{A}$ over a set $\mathcal{P}$ of attributes and associated with a linear secret sharing scheme (L, π). Let L be an $\ell \times m$ matrix. First, it picks random $\mathbf{s} = (s, s_2, \ldots, s_m) \xleftarrow{\$} \mathbb{Z}_p^m$ and computes share of s for $\pi(i)$ by $\lambda_i = \langle \mathbf{L}_i \cdot \mathbf{s} \rangle$ for $i = 1, \ldots, \ell$. It then computes $C_0 = \mathsf{M} \cdot e(g,g)^{\alpha \cdot s}$, $C_1 = g^s$. It also computes $(C_{i,1}, C_{i,2}, C_{i,3})$ for every $i = 1, \ldots, \ell$ as follows.

$$\begin{cases} C_{i,1} = W^{\lambda_i} V^{t_i},\ C_{i,2} = (U^{x_i} H)^{-t_i},\ C_{i,3} = g^{t_i} & \text{if } \pi(i) = x_i \\ C_{i,1} = W^{\lambda_i} (V')^{t_i},\ C_{i,2} = (U^{x_i} H)^{-t_i},\ C_{i,3} = g^{t_i} & \text{if } \pi(i) = x'_i \end{cases}$$

where $t_i \xleftarrow{\$} \mathbb{Z}_p$. The final output is $C = (C_0, C_1, \{C_{i,1}, C_{i,2}, C_{i,3}\}_{i \in [\ell]})$.

$\mathsf{Decrypt}(\mathsf{mpk}, C, \omega, \mathsf{sk}_{\tilde{\mathbb{A}}})$: Assume first that the policy $\tilde{\mathbb{A}}$ is satisfied by the attribute set ω, so that decryption is possible. Since $\tilde{\mathbb{A}} = NM(\mathbb{A})$ for some access structure $\mathbb{A}$ associated with a linear secret sharing scheme (L, π), we have $\omega' = N(\omega) \in \mathbb{A}$ and we let $I = \{i | \pi(i) \in \omega'\}$. Since ω' is authorized in $\mathbb{A}$, the receiver can efficiently compute reconstruction coefficients $\{(i, \mu_i)\}_{i \in I} = \mathsf{Recon}_{L,\pi}(\omega')$ such that $\sum_{i \in I} \mu_i \lambda_i = s$. It parses $C = (C_0, C_1, \{C_{i,1}, C_{i,2}, C_{i,3}\}_{i \in [\ell]})$, $\mathsf{sk}_\omega = \big(D_1, D_2, \{K_{i,1}, K_{i,2}, K'_{i,1}, K'_{i,2}\}_{i \in [k]}\big)$ and computes $e(g,g)^{r \cdot \lambda_i}$ for each $i \in I$ as

$$\begin{cases} e(C_{i,1}, D_2) \cdot e(C_{i,2}, K_{\tau,2}) \cdot e(C_{i,3}, K_{\tau,1}) \to e(g,W)^{r\lambda_i} & \text{if } \pi(i) = x_i \\ e(C_{i,1}, D_2) \cdot \prod\limits_{j \in [k]} \big(e(C_{i,3}, K'_{j,1}) \cdot e(C_{i,2}, K'_{j,2})\big)^{\frac{1}{x_i - \omega_j}} = e(g,W)^{r\lambda_i} & \text{if } \pi(i) = x'_i \end{cases}$$

where τ is the index such that $\omega_\tau = x_i$. Such τ exists if $i \in I$ and $\pi(i)$ is non-negated attribute. Next, it computes $e(C_1, D_1) \cdot \prod_{i \in I} \big(e(g,W)^{r\lambda_i}\big)^{-\mu_i} = e(g^s, g^\alpha) e(g,W)^{sr} e(g,W)^{-r \sum_{i \in I} \mu_i \lambda_i} = e(g,g)^{\alpha \cdot s}$. Finally, it recovers the message by $C_0 / e(g,g)^{s\alpha} = \mathsf{M}$.

The correctness of the scheme is proven in Appendix D. We can prove selective security of the scheme under the new assumption that we call n-(B) assumption which is secure in the generic group model. The definition of the assumption and the proof appear in Appendix F.

8 Comparisons

Here, we compare our schemes with existing schemes. In Table 1, we compare non-monotonic KP-ABE schemes with compact ciphertexts. In Table 2, we compare non-monotonic KP-ABE schemes from the DBDH assumption. In Table 3 (resp., 4), we compare the KP (resp., CP)-ABE schemes which allow unbounded size for set of attributes associated with ciphertext (resp., private key). In these tables, $\bar{n} = |\text{attribute set}| = |\omega|$, n is the maximum bound of $\bar{n}$ (i.e., $|\omega| < n$), φ is the number of allowed repetition of the same attributes which appear in a policy, and t_1 and t_2 are the number of

non-negated and negated attributes that apper in an access policy. We also let $t = t_1 + t_2$. The terms "reg-exp." and "mult-exp." refer to regular and multi-exponentiation in $\mathbb{G}$ and $\mathbb{G}_T$. The Pippenger algorithm [29] can efficiently compute the latter. The term "pair" refers to pairing computation. The columun "unbounded set" in Table 3 (resp., 4) states whether unbounded attribute set size is allowed for ciphertext (resp., for key) or not. The columun "unbounded multi-use" states whether unboudned reuse of the same policy for a key (resp., ciphertext) is allowed or not.

In Table 2, we only highlight the encryption cost. As for the efficiency of the decryption algorithm, our scheme in Sec. 5 is somewhat slower than [28], because of the additional exponentiations. Note that the schemes in [27] achieve adaptive security, whereas all the other schemes achieve only selective security.

Table 1: Comparison of non-monotonic KP-ABE with compact ciphertexts

Schemes	Master public key size ($\lvert\mathbb{G}\rvert, \lvert\mathbb{G}_T\rvert$)	Ciphertext overhead $\lvert\mathbb{G}\rvert$	Private key size $\lvert\mathbb{G}\rvert$	Computational cost for encryption (reg,mult)-exp	decryption (pair,mult-exp)	Assumption
ALP [3]	$(2n+2, 1)$	3	$(n+1)t$	$(2, 2)$	$(3, 3^*)$	n-DBDHE
Ours in Sec. 4.	$(n+2, 1)$	2	$(n+1)t + t_2$	$(2, 1)$	$(2, 2^*)$	n-DBDHE

* These multi-exponentiation is heavier than that needed in the encryption algorithm.

Table 2: Comparison of non-monotonic KP-ABE schemes from the DBDH

Schemes	Master public key size ($\lvert\mathbb{G}\rvert, \lvert\mathbb{G}_T\rvert$)	Ciphertext overhead $\lvert\mathbb{G}\rvert$	Private key size $\lvert\mathbb{G}\rvert$	Encryption cost reg-exp.	mult-exp.
OSW [28]	$(2n+2, 0)$	$2n-1$	$2t_1 + 3t_2$	2	$2n^\ddagger$
Ours in Sec. 5	$(n+2, 1)$	$n+1$	$2t_1 + 3t_2$	2	n

† For simplicity, we compare these schemes in a most basic form. However, we can modify the schemes so that the ciphertext size only depends on $\bar{n}$ instead of n, which might be preferable in many case, by the technique in [28]. As a result, master public key and the private key becomes larger, whereas it makes ciphertext size smaller and encryption/decryption cost lower.

‡ These multi-exponentiations are heavier than that of our scheme in Sec. 5.

Table 3: Comparison of KP-ABE schemes with unbounded attribute set size

Schemes	Access structure	Ciphertext overhead ($\lvert\mathbb{G}\rvert$)	unbounded set	Private key size($\mathbb{G}$)	unbounded multi-use	Assumption
LSW[21]	non-monotone	$3\bar{n}+1$	Yes	$2t + t_2$	Yes	RO+n-MEBDH
OT[27]	non-monotone	$14\bar{n}\varphi + 5$	Yes	$14t + 5$	No	DLIN
RW[30]	monotone	$2\bar{n} + 1$	Yes	$3t_1$	Yes	n-1assumption
LW[23]	monotone	$3\bar{n} + 1$	Yes	$4t_1$	Yes	assumption 1-4
Ours in Sec. 6	non-monotone	$4\bar{n} + 1$	Yes	$3t$	Yes	n-(A) assumption

§ LW scheme [23] is constructed in composite order group.

Table 4: Comparison of CP-ABE schemes with unbounded attribute set size

Schemes	Access structure	Ciphertext overhead ($\lvert\mathbb{G}\rvert$)	unbounded multi-use	Private key size($\mathbb{G}$)	unbounded set	Assumption
OT[27]	non-monotone	$14t + 5$	No	$14\bar{n}\varphi + 5$	Yes	DLIN
RW[30]	monotone	$3t_1 + 1$	Yes	$2\bar{n} + 2$	Yes	n-2 assumption
Ours in Sec. 7	non-monotone	$3t + 1$	Yes	$4\bar{n} + 2$	Yes	n-(B) assumption

References

[1] Nuttapong Attrapadung and Hideki Imai. Conjunctive broadcast and attribute-based encryption. In *Pairing*, pp. 248–265, 2009.

[2] Nuttapong Attrapadung and Benoît Libert. Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation. In *Public Key Cryptography*, pp. 384–402, 2010.

[3] Nuttapong Attrapadung, Benoît Libert, and Elie de Panafieu. Expressive key-policy attribute-based encryption with constant-size ciphertexts. In *Public Key Cryptography*, pp. 90–108, 2011.

[4] Amos Beimel. *Secure Schemes for Secret Sharing and Key Distribution*. PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1986.

[5] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*, pp. 321–334, 2007.

[6] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In *CRYPTO*, pp. 443–459, 2004.

[7] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *CRYPTO*, pp. 258–275, 2005.

[8] Dan Boneh and Michael Hamburg. Generalized identity based and broadcast encryption schemes. In *ASIACRYPT*, pp. 455–470, 2008.

[9] Dan Boneh, Amit Sahai, and Brent Waters. Functional Encryption: Definitions and Challenges. In *TCC*, pp. 253–273, 2011.

[10] Melissa Chase. Multi-authority attribute based encryption. In *TCC*, pp. 515–534, 2007.

[11] Melissa Chase and Sherman S. M. Chow. Improving privacy and security in multi-authority attribute-based encryption. In *ACM Conference on Computer and Communications Security*, pp. 121–130, 2009.

[12] Ling Cheung and Calvin C. Newport. Provably secure ciphertext policy abe. In *ACM Conference on Computer and Communications Security*, pp. 456–465, 2007.

[13] Sanjam Garg, Craig Gentry, Shai Halevi, Amit Sahai, and Brent Waters. Attribute-based encryption for circuits from multilinear maps. In *CRYPTO (2)*, pp. 479–499, 2013.

[14] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In *STOC*, pp. 545–554, 2013.

[15] Vipul Goyal, Abhishek Jain, Omkant Pandey, and Amit Sahai. Bounded ciphertext policy attribute based encryption. In *ICALP (2)*, pp. 579–591, 2008.

[16] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM Conference on Computer and Communications Security*, pp. 89–98, 2006.

[17] Susan Hohenberger and Brent Waters. Attribute-based encryption with fast decryption. In *Public Key Cryptography*, pp. 162–179, 2013.

[18] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT*, pp. 146–162, 2008.

[19] Allison B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In *EUROCRYPT*, pp. 318–335, 2012.

[20] Allison B. Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *EUROCRYPT*, pp. 62–91, 2010.

[21] Allison B. Lewko, Amit Sahai, and Brent Waters. Revocation systems with very small private keys. In *IEEE Symposium on Security and Privacy*, pp. 273–285, 2010.

[22] Allison B. Lewko and Brent Waters. Decentralizing attribute-based encryption. In *EUROCRYPT*, pp. 568–588, 2011.

[23] Allison B. Lewko and Brent Waters. Unbounded hibe and attribute-based encryption. In *EUROCRYPT*, pp. 547–567, 2011.

[24] Allison B. Lewko and Brent Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In *CRYPTO*, pp. 180–198, 2012.

[25] Moni Naor and Benny Pinkas. Efficient trace and revoke schemes. In *Financial Cryptography*, pp. 1–20, 2000.

[26] Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In *CRYPTO*, pp. 191–208, 2010.

[27] Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure unbounded inner-product and attribute-based encryption. In *ASIACRYPT*, pp. 349–366, 2012.

[28] Rafail Ostrovsky, Amit Sahai, and Brent Waters. Attribute-based encryption with non-monotonic access structures. In *ACM Conference on Computer and Communications Security*, pp. 195–203, 2007.

[29] Nicholas Pippenger. On the evaluation of powers and related problems (preliminary version). In *FOCS*, pp. 258–263, 1976.

[30] Yannis Rouselakis and Brent Waters. Practical constructions and new proof methods for large universe attribute-based encryption. *ACM Conference on Computer and Communications Security*, pp. 463–474, 2013.

[31] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, pp. 457–473, 2005.

[32] Brent Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *Public Key Cryptography*, pp. 53–70, 2011.

[33] Brent Waters. Functional encryption for regular languages. In *CRYPTO*, pp. 218–235, 2012.

A Proof of Lemmas

Proof. (of Lemma 1.) We define $\mathbf{h}'$ as $\mathbf{h}' = (h_n, \ldots, h_1)$. It holds that for all $\mathbf{z} \in \mathbb{Z}_p^n$, the coefficient of a^{n+1} in $\langle \mathbf{a}, \mathbf{h}' \rangle \langle \mathbf{a}, \mathbf{z} \rangle$ seen as a polynomial in a is $\langle \mathbf{h}, \mathbf{z} \rangle$.

BHSim samples $r' \xleftarrow{\$} \mathbb{Z}_p$ and implicitly sets $r = r' - \frac{\langle \mathbf{a}, \mathbf{h}' \rangle}{\langle \mathbf{h}, \mathbf{z}_0 \rangle}$. r is distributed uniformly random over $\mathbb{Z}_p$ as desired. In the following, we show that it is possible to efficiently compute $(g^{a^{n+1}+\alpha'} \cdot (g^{\langle \mathbf{z}_0, \mathbf{a} \rangle + w_0})^r, \{(g^{\langle \mathbf{z}_i, \mathbf{a} \rangle + w_i})^r\}_{i=1}^m)$.

We first show that $(g^{\langle \mathbf{z}_i, \mathbf{a} \rangle + w_i})^r$ can be computed efficiently for $i \in [m]$. To see this, it is enough to show that

$$r(\langle \mathbf{z}_i, \mathbf{a} \rangle + w_i) = r' \langle \mathbf{a}, \mathbf{z}_i \rangle - \frac{\langle \mathbf{a}, \mathbf{h}' \rangle \langle \mathbf{a}, \mathbf{z}_i \rangle}{\langle \mathbf{h}, \mathbf{z}_0 \rangle} + w_i r' - \frac{w_i \langle \mathbf{a}, \mathbf{h}' \rangle}{\langle \mathbf{h}, \mathbf{z}_0 \rangle}$$

is a polynomial in a with degree at most $2n$ and the coefficient of a^{n+1} is 0. It is straightforward to check that the degree of the polynomial is at most $2n$. We can also see that the coefficient of a^{n+1} is $-\frac{\langle \mathbf{h}, \mathbf{z}_i \rangle}{\langle \mathbf{h}, \mathbf{z}_0 \rangle} = 0$. We next show that $g^{a^{n+1}+\tilde{\alpha}} \cdot (g^{\langle \mathbf{z}_0, \mathbf{a} \rangle + w_0})^r$ can be computed efficiently. To see this, it is enough to show that

$$a^{n+1} + \tilde{\alpha} + r(\langle \mathbf{z}_0, \mathbf{a} \rangle + w_0) = a^{n+1} + \tilde{\alpha} + r' \langle \mathbf{z}_0, \mathbf{a} \rangle - \frac{\langle \mathbf{a}, \mathbf{h}' \rangle \langle \mathbf{a}, \mathbf{z}_0 \rangle}{\langle \mathbf{h}, \mathbf{z}_0 \rangle} + w_0 r' - \frac{w_0 \langle \mathbf{a}, \mathbf{h}' \rangle}{\langle \mathbf{h}, \mathbf{z}_0 \rangle}$$

is a polynomial in a with degree at most $2n$ and the coefficient of a^{n+1} is 0. It is straightforward to check that the degree of the polynomial is at most $2n$. We can also see that the coefficient of a^{n+1} is $1 - \frac{\langle \mathbf{h}, \mathbf{z}_0 \rangle}{\langle \mathbf{h}, \mathbf{z}_0 \rangle} = 0$. □

Proof. (of Lemma 2.) We define BBSim as follows: It chooses $r' \xleftarrow{\$} \mathbb{Z}_p$, and outputs $(A^{zr'} B^{-\frac{w}{z}} g^{wr'}, B^{-\frac{1}{z}} g^{r'})$. We claim that the output is correctly distributed. To see this, let $r = r' - \frac{b}{z}$. We can see that $B^{-\frac{1}{z}} g^{r'} = g^r$ and

$$A^{zr'} \cdot B^{-\frac{w}{z}} g^{wr'} = g^{ab + (az+w) \cdot (-\frac{b}{z} + r')} = g^{ab} \cdot (A^z g^w)^r$$

holds as desired. □

B Proof of Theorem 1

Proof. We construct an adversary $\mathcal{B}$ against selective security of TIBBE Π_{IBBE} scheme assuming that an adversary $\mathcal{A}$ against selective security of the KP-ABE system has non-negligible advantage.

Setup of master public key. At the outset of the game, the adversary $\mathcal{A}$ declares attribute set $\omega^\star$ that it intends to attack. Then, the TIBBE adveraary $\mathcal{B}$ announces $S^\star = \omega^\star$ as its challenge set. The master public key mpk is generated by the challenger and given to $\mathcal{B}$. $\mathcal{B}$ gives mpk to $\mathcal{A}$ as the master public key of the KP-ABE scheme.

Phase1 and 2. Throughout the game, $\mathcal{A}$ may ask for a private key of any access structure $\tilde{\mathbb{A}}$ such that $\omega^\star \notin \tilde{\mathbb{A}}$. By assumption, $\tilde{\mathbb{A}} = NM(\mathbb{A})$ for some monotonic access structure $\mathbb{A}$, defined over a set of parties, associated with a liner secret sharing scheme $\Pi = (L, \pi)$. Let L be an $\ell \times m$ matrix. Since $\omega^\star \notin \tilde{\mathbb{A}} = NM(\mathbb{A})$, we have $\omega' = N(\omega) \notin \mathbb{A}$. Therefore $\mathbf{1} = (1, 0, \ldots, 0)$ does not lie in the row space of $L_{\omega'}$, which is the submatrix of L formed by rows corresponding to attributes in ω'. Hence, due to the proposition 11 in [16], we have that there must exists an efficiently computable vector $\mathbf{z} \in \mathbb{Z}_p^m$ such that $\langle \mathbf{1}, \mathbf{z} \rangle = 1$ and $L_{\omega'} \cdot \mathbf{z}^\top = \mathbf{0}$. Now $\mathcal{B}$ picks $w_2, \ldots, w_m \xleftarrow{\$} \mathbb{Z}_p$ and implicitly defines $\mathbf{v} = (v_1, \ldots, v_m) = \alpha \mathbf{z} + \mathbf{w}$ where $\mathbf{w} = (0, w_2, \ldots, w_m)$. Note that we have that $v_1 = \alpha$ and that $v_2, \ldots, v_m \in \mathbb{Z}_p$ are uniformly distributed, as required in Definition 2. Next $\mathcal{B}$ implicitly defines each share of α as $\lambda_i = \langle \mathbf{L}_i, \mathbf{v} \rangle$, corresponding to a party named $\pi(i) = \breve{x}_i \in \mathcal{P}$ where x_i is the underlying

attribute ($\breve{x}_i$ being primed or unprimed). Although $\mathcal{B}$ cannot compute $\lambda_i = \langle \mathbf{L}_i, \mathbf{v} \rangle$ for all $i \in [\ell]$, it can compute $D_i = (d'_{i,1}, d'_{i,2})$ for all $i \in [\ell]$ as follows.

- For non-negated parties $\pi(i) = x_i$, $\mathcal{B}$ proceeds as follows.
  - If $x_i \in \omega^\star$, $\lambda_i = \langle \mathbf{L}_i, \mathbf{v} \rangle = \langle \mathbf{L}_i, \mathbf{w} \rangle$ holds since $\langle \mathbf{L}_i, \mathbf{z} \rangle = 0$ and thus λ_i can be efficiently computed by $\mathcal{B}$. In this case, $\mathcal{B}$ picks $r_i \xleftarrow{\$} \mathbb{Z}_p$ and computes
  $$D_i = (d'_{i,1}, d'_{i,2}) = \Big(g^{\lambda_i} \cdot \big(f_1^{\mathsf{IBBE}}(x_i)\big)^{r_i}, \big(f_2^{\mathsf{IBBE}}(x_i)\big)^{r_i} \Big).$$
  - If $x_i \notin \omega^\star$, $\mathcal{B}$ is allowed to query its challenger to extract $(d_1, d_2) \leftarrow \mathsf{KeyGen}'(\mathsf{msk}, (x_i, \mathsf{IBBE}))$. Also, we have $\lambda_i = \langle \mathbf{L}_i, \mathbf{v} \rangle = \mu_1 \cdot \alpha + \mu_2$ where the coefficients $\mu_1 = \langle \mathbf{L}_i, \mathbf{z} \rangle$ and $\mu_2 = \langle \mathbf{L}_i, \mathbf{w} \rangle$ are both efficiently computable. $\mathcal{B}$ can compute well formed $D_i = (d'_{i,1}, d'_{i,2})$ by setting
  $$D_i = (d_1^{\mu_1} \cdot g^{\mu_2} \cdot \big(f_1^{\mathsf{IBBE}}(x_i)\big)^{\tilde{r}_i}, d_2^{\mu_1} \cdot \big(f_2^{\mathsf{IBBE}}(x_i)\big)^{\tilde{r}_i})$$
  where $\tilde{r}_i \xleftarrow{\$} \mathbb{Z}_p$.
- For negated parties $\pi(i) = x'_i$, $\mathcal{B}$ proceeds as follows.
  - If $x_i \in \omega^\star$, $\mathcal{B}$ is allowed to query its challenger to extract $(d_1, d_2) \leftarrow \mathsf{KeyGen}'(\mathsf{msk}, (x_i, \mathsf{IBR}))$. Also, we have $\lambda_i = \langle \mathbf{L}_i, \mathbf{v} \rangle = \mu_1 \cdot \alpha + \mu_2$ where the coefficients $\mu_1 = \langle \mathbf{L}_i, \mathbf{z} \rangle$ and $\mu_2 = \langle \mathbf{L}_i, \mathbf{w} \rangle$ are both efficiently computable. $\mathcal{B}$ can compute well formed $D_i = (d'_{i,1}, d'_{i,2})$ by setting
  $$D_i = (d_1^{\mu_1} \cdot g^{\mu_2} \cdot \big(f_1^{\mathsf{IBR}}(x_i)\big)^{\tilde{r}_i}, d_2^{\mu_1} \cdot \big(f_2^{\mathsf{IBR}}(x_i)\big)^{\tilde{r}_i})$$
  where $\tilde{r}_i \xleftarrow{\$} \mathbb{Z}_p$.
  - If $x_i \notin \omega^\star$, $\lambda_i = \langle \mathbf{L}_i, \mathbf{v} \rangle = \langle \mathbf{L}_i, \mathbf{w} \rangle$ holds since $\langle \mathbf{L}_i, \mathbf{z} \rangle = 0$ and thus λ_i can be efficiently computed by $\mathcal{B}$. In this case, $\mathcal{B}$ picks $r_i \xleftarrow{\$} \mathbb{Z}_p$ and computes
  $$D_i = (d'_{i,1}, d'_{i,2}) = \Big(g^{\lambda_i} \cdot \big(f_1^{\mathsf{IBR}}(x_i)\big)^{r_i}, \big(f_2^{\mathsf{IBR}}(x_i)\big)^{r_i} \Big).$$

Finally, $\mathcal{B}$ returns private key $\mathsf{sk}_{\tilde{\mathbb{A}}} = \{D_i\}_{i=1}^{\ell}$ to $\mathcal{A}$.
Challenge. At some point in the game, $\mathcal{A}$ submits a pair of ciphertexts $(\mathsf{M}_0, \mathsf{M}_1)$ to $\mathcal{B}$. Then $\mathcal{B}$ declares the same message for its challenger. $\mathcal{B}$ is given a challenge ciphertext and relays it to $\mathcal{A}$.
Guess. Finally, $\mathcal{A}$ outputs its guess β' for β. Then $\mathcal{B}$ outputs β' as its guess. It is easy to see that $\mathcal{B}$ is successful whenever $\mathcal{A}$ is so, and thus $\mathcal{B}$ has the same advantage as $\mathcal{A}$. □

C Concrete Descriptions of Our KP-ABE Schemes

Here, we give concrete descriptions of our KP-ABE schemes obtained by applying our conversion in section 3 to our TIPE schemes in Sec. 4, 5 and TIBBE scheme in Sec. 6.

C.1 The KP-ABE Scheme Derived from the TIPE Scheme in Section 4

We can derive a TIBBE scheme from the TIPE scheme proposed in Sec. 4 as we explained in Sec. 3.1. Then we can apply the conversion in Sec. 3 to this TIBBE, by setting $(f_1^{\mathsf{IBBE}}, f_2^{\mathsf{IBBE}}, f_1^{\mathsf{IBR}}, f_2^{\mathsf{IBR}}, F)$ as

$$f_1^{\mathsf{IBBE}}(\mathsf{ID}) \;= V, \quad f_2^{\mathsf{IBBE}}(\mathsf{ID}) \;= \big(g,\ \{U_1^{-\frac{x_i}{x_1}} U_i\}_{i=2\ldots,n}\big),$$

$$f_1^{\mathsf{IBR}}(\mathsf{ID}) = U_1, \quad f_2^{\mathsf{IBR}}(\mathsf{ID}) = \left(g,\ V,\ \{U_1^{-\frac{x_i}{x_1}} U_i\}_{i=2,\ldots,n}\right)$$
$$F(\mathsf{ID}_1, \ldots, \mathsf{ID}_k) = \left(g^s, (VU_1^{y_1} \cdots U_n^{y_n})^{-s}\right)$$

where $\mathbf{x} = (x_1, \ldots, x_n) = (1, \mathsf{ID}, \ldots, \mathsf{ID}^{n-1})$ and $\mathbf{y} = (y_1, \ldots, y_n)$ is a vector whose first $k+1$ coordinates are the coefficients of the polynomial $P[Z] = \sum_{i=1}^{k+1} y_i Z^{i-1} = \prod_{j \in [k]} (Z - \mathsf{ID}_j)$. As a result, we obtain a new non-monotonic KP-ABE scheme with very short ciphertext. We give the concrete description of the resulting scheme below.

$\mathsf{Setup}(\lambda, n)$: It chooses bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ with $g \xleftarrow{\$} \mathbb{G}$. It also picks $v, \alpha \xleftarrow{\$} \mathbb{Z}_p$ and $\mathbf{u} = (u_1, \ldots, u_n) \xleftarrow{\$} \mathbb{Z}_p^n$. Then it sets $V = g^v$ and $U = (U_1, \ldots, U_n) = g^{\mathbf{u}}$. It finally outputs the master public key $\mathsf{mpk} = (g, U_1, \ldots, U_n, V, e(g,g)^\alpha)$ and the master secret key $\mathsf{msk} = \alpha$.

$\mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, \tilde{\mathbb{A}})$: The input to the algorithm is the master secret key msk, the master public key mpk, and a non-monotonic access structure $\tilde{\mathbb{A}}$ such that we have $\tilde{\mathbb{A}} = NM(\mathbb{A})$ for some monotonic access structure $\mathbb{A}$ over a set $\mathcal{P}$ of attributes and associated with a linear secret sharing scheme (L, π). Let L be an $\ell \times m$ matrix. First, it generates shares of α with (L, π). Namely, it chooses a vector $\mathbf{s} = (\alpha, s_2, \ldots, s_m)$ where $s_2, \ldots, s_m \xleftarrow{\$} \mathbb{Z}_p$ and calculates $\lambda_i = \langle \mathbf{L}_i, \mathbf{s} \rangle$ for each $i = 1, \ldots, \ell$. The party corresponds to share λ_i is $\pi(i) \in \mathcal{P}$, where $\pi(i) = x_i$ (i.e., non-negated) or $\pi(i) = x_i'$ (negated). Then for each $i = 1, \ldots, \ell$, it picks $r_i \xleftarrow{\$} \mathbb{Z}_p$ and defines $\boldsymbol{\rho}_i = (\rho_{i,1}, \ldots, \rho_{i,n}) = (1, x_i, x_i^2, \ldots, x_i^{n-1})$. That is $\rho_{i,j} = x_i^{j-1}$. It computes D_i for each $i = 1, \ldots, \ell$ as follows.

- If $\pi(i) = x_i$, it computes

$$D_i = (D_{i,1}^{(1)}, D_{i,2}^{(1)}, \{K_{i,j}^{(1)}\}_{j=2}^n) = \left(g^{\lambda_i} \cdot V^{r_i},\ g^{r_i},\ \{\left(U_1^{-\frac{\rho_{i,j}}{\rho_{i,1}}} \cdot U_j\right)^{r_i}\}_{j=2}^n\right).$$

- If $\pi(i) = x_i'$ where x_i is underlying attribute, it computes

$$D_i = (D_{i,1}^{(2)}, D_{i,2}^{(2)}, D_{i,3}^{(2)}, \{K_{i,j}^{(2)}\}_{j=2}^n) = \left(g^{\lambda_i} \cdot U_1^{r_i},\ g^{r_i},\ V^{r_i},\ \{(U_1^{-\frac{\rho_{i,j}}{\rho_{i,1}}} U_j)^{r_i}\}_{j=2}^n\right).$$

It then outputs the private key as $\mathsf{sk}_{\tilde{\mathbb{A}}} = \{D_i\}_{i=1}^\ell$

$\mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}, \omega)$: To encrypt $\mathsf{M} \in \mathbb{G}_T$ for the set of attributes ω (with $|\omega| < n$), it first defines $\mathbf{y} = (y_1, \ldots, y_n)$ as the vector whose first $|\omega|+1$ coordinates are the coefficients of the polynomial $P_\omega[Z] = \sum_{i=1}^{|\omega|+1} y_i Z^{i-1} = \prod_{j \in \omega} (Z - j)$. If $|\omega| + 1 < n$, set $y_j = 0$ for $|\omega| + 2 \le j \le n$. Then it picks $s \xleftarrow{\$} \mathbb{Z}_p$ and computes the ciphertext as

$$C = \left(C_0 = \mathsf{M} \cdot e(g,g)^{\alpha s}, C_1 = g^s, C_2 = (VU_1^{y_1} \cdots U_n^{y_n})^{-s}\right).$$

$\mathsf{Decrypt}(\mathsf{mpk}, C, \omega, \mathsf{sk}_{\tilde{\mathbb{A}}})$**:** Assume first that the policy $\tilde{\mathbb{A}}$ is satisfied by the attribute set ω, so that decryption is possible. Since $\tilde{\mathbb{A}} = NM(\mathbb{A})$ for some access structure $\mathbb{A}$ associated with a linear secret sharing scheme (L, π), we have $\omega' = N(\omega) \in \mathbb{A}$ and we let $I = \{i | \pi(i) \in \omega'\}$. Since ω' is authorized in $\mathbb{A}$, the receiver can efficiently compute reconstruction coefficients $\{(i, \mu_i)\}_{i \in I} = \mathsf{Recon}_{L,\pi}(\omega')$ such that $\sum_{i \in I} \mu_i \lambda_i = \alpha$ (although the shares are not known to the receiver). Let $\mathbf{y} = (y_1, \ldots, y_n)$ be the vector containing the coefficients of the polynomial

$P_\omega[Z] = \prod_{j\in\omega}(Z-j) = \sum_{i=1}^{|\omega|+1} y_i Z^{i-1}$. Then, it parses $C = (C_0, C_1, C_2)$, $\mathsf{sk}_{\tilde{\mathbb{A}}} = \{D_i\}_{i=1}^{\ell}$ and retrieves $e(g,g)^{\lambda_i s}$ for each $i \in I$ as follows.

$$\begin{cases} e(C_1, D_{i,1}^{(1)} \cdot \prod_{j=2}^{n} K_{i,j}^{(1)y_j}) \cdot e(C_2, D_{i,2}^{(1)}) = e(g,g)^{s\lambda_i} & \text{if } \pi(i) = x_i \\ e(C_1, D_{i,1}^{(2)}) \cdot \Big(e(C_1, D_{i,3}^{(2)} \prod_{j=2}^{n} K_{i,j}^{(2)y_j}) \cdot e(C_2, D_{i,2}^{(2)})\Big)^{\frac{\rho_{i,1}}{\langle \boldsymbol{\rho}_i, \mathbf{y}\rangle}} = e(g,g)^{s\lambda_i} & \text{if } \pi(i) = x'_i. \end{cases}$$

Finally, it recovers the message by $\mathsf{M} = C_0 \cdot \prod_{i\in I}\big(e(g,g)^{\lambda_i s}\big)^{-\mu_i}$.

If we split I into $I_0 \cup I_1$, where I_0 and I_1 correspond to unprimed and primed attributes, respectively, decryption can more efficiently compute

$$e\Big(C_1, \prod_{i\in I_0}(D_{i,1}^{(1)\mu_i} K_i^{(1)\mu_i}) \cdot \prod_{i\in I_1} D_{i,1}^{(2)\mu_i} \cdot \big(D_{i,3}^{(2)} \cdot K_i^{(2)}\big)^{\frac{\mu_i\cdot\rho_{i,1}}{\langle \boldsymbol{\rho}_i, \mathbf{y}\rangle}}\Big) \cdot e\Big(C_2, \prod_{i\in I_0} D_{i,2}^{(1)\mu_i} \cdot \prod_{i\in I_1} D_{i,2}^{(2)\frac{\mu_i\cdot\rho_{i,1}}{\langle \boldsymbol{\rho}_i, \mathbf{y}\rangle}}\Big) = e(g,g)^{s\alpha}$$

where $K_i^{(1)} = \prod_{j=2}^{n} K_{i,j}^{(1)y_j}, K_i^{(2)} = \prod_{j=2}^{n} K_{i,j}^{(2)y_j}$ in the above. Thus, we need only two pairing evaluations in the decryption algorithm.

C.2 The KP-ABE Scheme Derived from the TIPE Scheme in Section 5

We can derive a TIBBE scheme from the TIPE scheme proposed in Sec. 5 as we explained in Sec. 3.1. Then we can apply the conversion in Sec. 3 to this TIBBE, by setting $(f_1^{\mathsf{IBBE}}, f_2^{\mathsf{IBBE}}, f_1^{\mathsf{IBR}}, f_2^{\mathsf{IBR}}, F)$ as

$$\begin{aligned} f_1^{\mathsf{IBBE}}(\mathsf{ID}) &= V_1^{x_1} \cdots V_n^{x_n}, \quad f_2^{\mathsf{IBBE}}(\mathsf{ID}) = g, \\ f_1^{\mathsf{IBR}}(\mathsf{ID}) &= U, \quad f_2^{\mathsf{IBR}}(\mathsf{ID}) = \big(V_1^{x_1} \cdots V_n^{x_n}, g\big) \\ F(\mathsf{ID}_1, \ldots, \mathsf{ID}_k) &= \big(g^s, \{(U^{y_i} V_i)^{-s}\}_{i\in[n]}\big) \end{aligned}$$

where $\mathbf{x} = (x_1, \ldots, x_n) = (1, \mathsf{ID}, \ldots, \mathsf{ID}^{n-1})$ and $\mathbf{y} = (y_1, \ldots, y_n)$ is a vector whose first $k+1$ coordinates are the coefficients of the polynomial $P[Z] = \sum_{i=1}^{k+1} y_i Z^{i-1} = \prod_{j\in[k]}(Z - \mathsf{ID}_j)$. As a result, we obtain a new non-monotonic KP-ABE scheme from the DBDH assumption. We give the concrete description of the resulting scheme below.

$\mathsf{Setup}(\lambda, n)$: It chooses bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ with $g \xleftarrow{\$} \mathbb{G}$. It also picks $u, \alpha \xleftarrow{\$} \mathbb{Z}_p$ and $\mathbf{v} = (v_1, \ldots, v_n) \xleftarrow{\$} \mathbb{Z}_p^n$. Then it sets $U = g^u$ and $\mathbf{V} = (V_1, \ldots, V_n) = g^{\mathbf{v}}$. It finally outputs the master public key $\mathsf{mpk} = (g, U, V_1, \ldots, V_n, e(g,g)^\alpha)$ and the master secret key $\mathsf{msk} = \alpha$.

$\mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}, \omega)$: To encrypt $\mathsf{M} \in \mathbb{G}_T$ for the set of attributes ω (with $|\omega| < n$), it first defines $\mathbf{y} = (y_1, \ldots, y_n)$ as the vector whose first $|\omega|+1$ coordinates are the coefficients of the polynomial $P_\omega[Z] = \sum_{i=1}^{|\omega|+1} y_i Z^{i-1} = \prod_{j\in\omega}(Z-j)$. If $|\omega| + 1 < n$, set $y_j = 0$ for $|\omega| + 2 \le j \le n$. Then it picks $s \xleftarrow{\$} \mathbb{Z}_p$ and computes the ciphertext as

$$C = \Big(C_0 = \mathsf{M} \cdot e(g,g)^{\alpha s},\ C_1 = g^s,\ \{E_i = (U^{y_i} V_i)^{-s}\}_{i=1,\ldots n}\Big).$$

$\mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, \tilde{\mathbb{A}})$: The input to the algorithm is the master secret key msk, the master public key mpk, and a non-monotonic access structure $\tilde{\mathbb{A}}$ such that we have $\tilde{\mathbb{A}} = NM(\mathbb{A})$ for some monotonic access structure $\mathbb{A}$ over a set $\mathcal{P}$ of attributes and associated with a linear secret sharing scheme (L, π). Let L be an $\ell \times m$ matrix. First, it generates shares of $\{\lambda_i\}_{i\in[\ell]}$ of α. The party corresponds to share λ_i is $\pi(i) \in \mathcal{P}$, where $\pi(i) = x_i$ (i.e., non-negated) or $\pi(i) = x_i'$ (negated). Then for each $i = 1, \ldots, \ell$, it picks $r_i \xleftarrow{\$} \mathbb{Z}_p$ and defines $\boldsymbol{\rho}_i = (\rho_{i,1}, \ldots, \rho_{i,n}) = (1, x_i, x_i^2, \ldots, x_i^{n-1})$. It computes D_i for each $i = 1, \ldots, \ell$ as follows.

$$D_i = \begin{cases} \left(D_{i,1}^{(1)} = g^{\lambda_i} \cdot (V_1^{\rho_{i,1}} \cdots V_n^{\rho_{i,n}})^{r_i},\ D_{i,2}^{(1)} = g^{r_i}\right) & \text{if } \pi(i) = x_i \\ \left(D_{i,1}^{(2)} = g^{\lambda_i} U^{r_i},\ D_{i,2}^{(2)} = (V_1^{\rho_{i,1}} \cdots V_n^{\rho_{i,n}})^{r_i},\ D_{i,3}^{(2)} = g^{r_i}\right) & \text{if } \pi(i) = x_i'. \end{cases}$$

It then outputs the private key as $\mathsf{sk}_{\tilde{\mathbb{A}}} = \{D_i\}_{i=1}^{\ell}$

$\mathsf{Decrypt}(\mathsf{mpk}, C, \omega, \mathsf{sk}_{\tilde{\mathbb{A}}})$**:** Assume first that the policy $\tilde{\mathbb{A}}$ is satisfied by the attribute set ω, so that decryption is possible. Since $\tilde{\mathbb{A}} = NM(\mathbb{A})$ for some access structure $\mathbb{A}$ associated with a linear secret sharing scheme (L, π), we have $\omega' = N(\omega) \in \mathbb{A}$ and we let $I = \{i | \pi(i) \in \omega'\}$. Since ω' is authorized in $\mathbb{A}$, the receiver can efficiently compute reconstruction coefficients $\{(i, \mu_i)\}_{i\in I} = \mathsf{Recon}_{L,\pi}(\omega')$ such that $\sum_{i\in I} \mu_i \lambda_i = \alpha$. Let $\mathbf{y} = (y_1, \ldots, y_n)$ be the vector containing the coefficients of the polynomial $P_\omega[Z] = \prod_{j\in\omega}(Z - j) = \sum_{i=1}^{|\omega|+1} y_i Z^{i-1}$. Then, it parses $C = (C_0, C_1, \{E_i\}_{i=1}^n)$, $\mathsf{sk}_{\tilde{\mathbb{A}}} = \{D_i\}_{i=1}^{\ell}$ and retrieve $e(g,g)^{\lambda_i s}$ for each $i \in I$ as follows.

$$\begin{cases} e(C_1, D_{i,1}^{(1)}) \cdot e(\prod_{j=1}^{n} E_j^{\rho_{i,j}}, D_{i,2}^{(1)}) = e(g,g)^{s\lambda_i} & \text{if } \pi(i) = x_i \\ e(C_1, D_{i,1}^{(2)}) \Big(e(\prod_{j=1}^{n} E_j^{\rho_{i,j}}, D_{i,3}^{(2)}) \cdot e(C_1, D_{i,2}^{(2)}) \Big)^{\frac{1}{\langle \boldsymbol{\rho}_i, \mathbf{y}\rangle}} = e(g,g)^{s\lambda_i} & \text{if } \pi(i) = x_i'. \end{cases}$$

Finally, it recovers the message by $\mathsf{M} = C_0 \cdot \prod_{i\in I} \big(e(g,g)^{\lambda_i s}\big)^{-\mu_i}$.

C.3 The KP-ABE scheme Derived from the TIBBE scheme in Section 6

We can apply the conversion in Sec. 3 to a TIBBE scheme in Sec. 6, by setting $(f_1^{\mathsf{IBBE}}, f_2^{\mathsf{IBBE}}, f_1^{\mathsf{IBR}}, f_2^{\mathsf{IBR}}, F)$ as

$$\begin{array}{rcl} f_1^{\mathsf{IBBE}}(\mathsf{ID}) & = & W, \quad f_2^{\mathsf{IBBE}}(\mathsf{ID}) = (U^{\mathsf{ID}} H,\ g) \\ f_1^{\mathsf{IBR}}(\mathsf{ID}) & = & B', \quad f_2^{\mathsf{IBR}}(\mathsf{ID}) = (B^{\mathsf{ID}} V,\ g) \end{array}$$

and

$$F(s, \mathsf{ID}_1, \ldots, \mathsf{ID}_k) = \left(C_1 = g^s, \left\{ \begin{array}{ll} C_{i,1} = W^{-s}(U^{\mathsf{ID}_i} H)^{-t_i}, & C_{i,2} = g^{t_i} \\ C'_{i,1} = (B'^{\mathsf{ID}_i} V')^{-s_i}, & C'_{i,2} = B^{s_i} \end{array} \right\}_{i\in[k]} \right).$$

where $t_1 \ldots, t_k \xleftarrow{\$} \mathbb{Z}_p$ and $s_1, \ldots, s_k$ are random elements in $\mathbb{Z}_p$ such that $s_1 + \cdots + s_k = s$. As a result, we obtain a first non-monotonic unbounded KP-ABE scheme in the standard model. We give the concrete description of the resulting scheme below.

$\mathsf{Setup}(\lambda)$: It chooses bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ with $g \xleftarrow{\$} \mathbb{G}$. It also picks $H, U, V, W \xleftarrow{\$} \mathbb{G}$ and $b, \alpha \xleftarrow{\$} \mathbb{Z}_p^n$. Then it sets $B = g^b, B' = g^{b^2}, V' = V^b$. It finally outputs the master public key $\mathsf{mpk} = (g, H, U, W, B, B', V, V', e(g,g)^\alpha)$ and the master secret key $\mathsf{msk} = \alpha$.

$\mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}, \omega)$: To encrypt $\mathsf{M} \in \mathbb{G}_T$ for the set of attributes $\omega = (\omega_1, \ldots, \omega_k)$, it chooses $s, t_1, \ldots, t_k \xleftarrow{\$} \mathbb{Z}_p$ and random $s_1, \ldots, s_k$ such that $s_1 + \ldots + s_k = s$ and computes the ciphertext as

$$C = \left(C_0 = \mathsf{M} \cdot e(g,g)^{\alpha s}, C_1 = g^s, \left\{ \begin{array}{ll} C_{i,1} = W^{-s}(U^{\omega_i} H)^{-t_i}, & C_{i,2} = g^{t_i} \\ C'_{i,1} = (B'^{\omega_i} V')^{-s_i}, & C'_{i,2} = B^{s_i} \end{array} \right\}_{i \in [k]} \right).$$

$\mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, \tilde{\mathbb{A}})$: The input to the algorithm is the master secret key msk, the master public key mpk, and a non-monotonic access structure $\tilde{\mathbb{A}}$ such that we have $\tilde{\mathbb{A}} = NM(\mathbb{A})$ for some monotonic access structure $\mathbb{A}$ over a set $\mathcal{P}$ of attributes and associated with a linear secret sharing scheme (L, π). Let L be an $\ell \times m$ matrix. First, it generates shares of $\{\lambda_i\}_{i \in [\ell]}$ of α. The party corresponds to share λ_i is $\pi(i) \in \mathcal{P}$, where $\pi(i) = x_i$ (i.e., non-negated) or $\pi(i) = x'_i$ (negated). Then for each $i = 1, \ldots, \ell$, it picks $r_i \xleftarrow{\$} \mathbb{Z}_p$ and computes D_i as follows.

$$D_i = \left\{ \begin{array}{ll} \left(D^{(1)}_{i,1} = g^{\lambda_i} \cdot W^{r_i}, D^{(1)}_{i,2} = (U^{x_i} H)^{r_i}, D^{(1)}_{i,3} = g^{r_i} \right) & \text{if } \pi(i) = x_i \\ \left(D^{(2)}_{i,1} = g^{\lambda_i} \cdot (B')^{r_i}, D^{(2)}_{i,2} = (B^{x_i} V)^{r_i}, D^{(2)}_{i,3} = g^{r_i} \right) & \text{if } \pi(i) = x'_i. \end{array} \right.$$

It then outputs the private key as $\mathsf{sk}_{\tilde{\mathbb{A}}} = \{D_i\}_{i=1}^\ell$

$\mathsf{Decrypt}(\mathsf{mpk}, C, \omega, \mathsf{sk}_{\tilde{\mathbb{A}}})$**:** Assume first that the policy $\tilde{\mathbb{A}}$ is satisfied by the attribute set ω, so that decryption is possible. Since $\tilde{\mathbb{A}} = NM(\mathbb{A})$ for some access structure $\mathbb{A}$ associated with a linear secret sharing scheme (L, π), we have $\omega' = N(\omega) \in \mathbb{A}$ and we let $I = \{i | \pi(i) \in \omega'\}$. Since ω' is authorized in $\mathbb{A}$, the receiver can efficiently compute reconstruction coefficients $\{(i, \mu_i)\}_{i \in I} = \mathsf{Recon}_{L,\pi}(\omega')$ such that $\sum_{i \in I} \mu_i \lambda_i = \alpha$.

Then, it parses $C = (C_0, C_1, \{E_i\}_{i=1}^n)$, $\mathsf{sk}_{\tilde{\mathbb{A}}} = \{D_i\}_{i=1}^\ell$ and retrieves $e(g,g)^{\lambda_i s}$ for each $i \in I$ as follows.

$$\left\{ \begin{array}{ll} e(C_1, D^{(1)}_{i,1}) \cdot e(C_{\tau,1}, D^{(1)}_{i,3}) \cdot e(C_{\tau,2}, D^{(1)}_{i,2}) = e(g,g)^{s\lambda_i} & \text{if } \pi(i) = x_i \\ e(C_1, D^{(2)}_{i,1}) \cdot \prod_{j=1}^{k} \left(e(C'_{j,1}, D^{(2)}_{i,3}) \cdot e(C'_{j,2}, D^{(2)}_{i,2}) \right)^{1/(\omega_j - x_i)} = e(g,g)^{s\lambda_i} & \text{if } \pi(i) = x'_i. \end{array} \right.$$

In the above, τ is a index such that $\omega_\tau = x_i = \pi(i)$. Such τ exists if $i \in I$ and $\pi(i)$ is non-negated attribute. Finally, it recovers the message by $\mathsf{M} = C_0 \cdot \prod_{i \in I} \left(e(g,g)^{\lambda_i s} \right)^{-\mu_i}$.

D Proof of Correctness

Correctness of the Scheme in Section 4. In the case of $\langle \mathbf{x}, \mathbf{y} \rangle = 0$ and $\mathsf{type} = \mathsf{ZIPE}$, we have

$$D_1 \cdot \prod_{i=2}^{n} K_i^{y_i} = g^\alpha V^r \cdot \left(U_1^{\frac{x_1 y_1 - \langle \mathbf{x}, \mathbf{y} \rangle}{x_1}} \cdot \prod_{i=2}^{n} U_i^{y_i} \right)^r = g^\alpha \cdot (V \prod_{i=1}^{n} U_i^{y_i})^r$$

and thus

$$e(C_1, D_1 \cdot \prod_{i=2}^{n} K_i^{y_i}) \cdot e(C_2, D_2) = e(g^s, g^\alpha \cdot (V \cdot \prod_{i=1}^{n} U_i^{y_i})^r) \cdot e((V \cdot \prod_{i=1}^{n} U_i^{y_i})^{-s}, g^r) = e(g,g)^{s\alpha}.$$

In the case of $\langle \mathbf{x}, \mathbf{y} \rangle \neq 0$ and $\mathsf{type} = \mathsf{NIPE}$, we have

$$D_1 \cdot \prod_{i=2}^{n} K_i^{y_i} = V^r \cdot \big(U_1^{\frac{x_1 y_1 - \langle \mathbf{x}, \mathbf{y} \rangle}{x_1}} \cdot \prod_{i=2}^{n} U_i^{y_i}\big)^r = U_1^{\frac{-r\langle \mathbf{x}, \mathbf{y} \rangle}{x_1}} \cdot (V \prod_{i=1}^{n} U_i^{y_i})^r$$

and thus

$$\begin{aligned} & e(C_1, D_1) \cdot \Big(e(C_1, D_3 \prod_{i=2}^{n} K_i^{y_i}) \cdot e(C_2, D_2)\Big)^{\frac{x_1}{\langle \mathbf{x}, \mathbf{y} \rangle}} \\ = \; & e(g^s, g^\alpha \cdot U_1^r) \cdot \Big(e\big(g^s, U_1^{\frac{-r\langle \mathbf{x}, \mathbf{y} \rangle}{x_1}} \cdot (V \cdot \prod_{i=1}^{n} U_i^{y_i})^r\big) \cdot e\big((V \cdot \prod_{i=1}^{n} U_i^{y_i})^{-s}, g^r\big)\Big)^{\frac{x_1}{\langle \mathbf{x}, \mathbf{y} \rangle}} \\ = \; & e(g,g)^{\alpha \cdot s} \cdot e(g, U_1)^{sr} \cdot e(g, U_1)^{-sr} = e(g,g)^{\alpha \cdot s}. \end{aligned}$$

Correctness of the Scheme in Section 5. We first observe that $\prod_{i=1}^{n} E_i^{x_i} = \big(U^{\langle \mathbf{x}, \mathbf{y} \rangle} \cdot \prod_{i=1}^{n} V_i^{x_i}\big)^{-s}$. In the case of $\langle \mathbf{x}, \mathbf{y} \rangle = 0$ and $\mathsf{type} = \mathsf{ZIPE}$, we have

$$e(C_1, D_1) \cdot e(\prod_{i=1}^{n} E_i^{x_i}, D_2) = e\big(g^s, g^\alpha (\prod_{i=1}^{n} V_i^{x_i})^r\big) \cdot e\big((\prod_{i=1}^{n} V_i^{x_i})^{-s}, g^r\big) = e(g,g)^{s\alpha}.$$

In the case of $\langle \mathbf{x}, \mathbf{y} \rangle \neq 0$ and $\mathsf{type} = \mathsf{NIPE}$, we have

$$\begin{aligned} & e(C_1, D_1)\Big(e(\prod_{i=1}^{n} E_i^{x_i}, D_3) \cdot e(C_1, D_2)\Big)^{\frac{1}{\langle \mathbf{x}, \mathbf{y} \rangle}} \\ = \; & e\big(g^s, g^\alpha \cdot U^r\big) \cdot \Big(e\big((U^{\langle \mathbf{x}, \mathbf{y} \rangle} \cdot \prod_{i=1}^{n} V_i^{x_i})^{-s}, g^r\big) \cdot e(g^s, \prod_{i=1}^{n} V_i^{x_i})\big)\Big)^{\frac{1}{\langle \mathbf{x}, \mathbf{y} \rangle}} \\ = \; & e(g,g)^{s\alpha} \cdot e(g, U)^{sr} \cdot e(g, U)^{-sr} = e(g,g)^{s\alpha}. \end{aligned}$$

Correctness of the Scheme in Section 6. In the case of $\mathsf{ID} \in S$ and $\mathsf{type} = \mathsf{IBBE}$, we have

$$\begin{aligned} e(C_1, D_1) \cdot e(C_{\tau,1}, D_3) \cdot e(C_{\tau,2}, D_2) &= e(g^s, g^\alpha W^r) \cdot e(W^{-s} (U^{\mathsf{ID}_\tau} H)^{-t_\tau}, g^r) \cdot e(g^{t_\tau}, (U^{\mathsf{ID}_\tau} H)^r) \\ &= e(g,g)^{s\alpha} \cdot e(g, W)^{sr} \cdot e(g, W)^{-sr} = e(g,g)^{s\alpha}. \end{aligned}$$

In the case of $\mathsf{ID} \notin S$ and $\mathsf{type} = \mathsf{IBR}$, we have

$$\begin{aligned} & e(C_1, D_1) \cdot \prod_{i=1}^{k} \big(e(C'_{i,1}, D_3) \cdot e(C'_{i,2}, D_2)\big)^{1/(\mathsf{ID} - \mathsf{ID}_i)} \\ = \; & e(g^s, g^\alpha \cdot g^{b^2 r}) \prod_{i \in [k]} \Big(e\big((g^{b^2 \mathsf{ID}_i} V^b)^{-s_i}, g^r\big) \cdot e\big(g^{b s_i}, (g^{b \mathsf{ID}} V)^r\big)\Big)^{1/(\mathsf{ID}_i - \mathsf{ID})} \\ = \; & e(g,g)^{s\alpha} \cdot e(g,g)^{s b^2 r} \cdot \prod_{i \in [k]} e(g,g)^{-s_i b^2 r} = e(g,g)^{s\alpha}. \end{aligned}$$

Correctness of the Scheme in Section 7. It suffices to show that the decryption algorithm correctly recovers $e(g, W)^{r\lambda_i}$ for each row $i \in I$. In the case of $\pi(i) = x_i$, we have

$$\begin{aligned} e(C_{i,1}, D_2) \cdot e(C_{i,2}, K_{\tau,2}) \cdot e(C_{i,3}, K_{\tau,1}) &= e(W^{\lambda_i} V^{t_i}, g^r) \cdot e((U^{x_i} H)^{-t_i}, g^{r_\tau}) \cdot e(g^{t_i}, V^{-r} \cdot (U^{\omega_\tau} H)^{-r_\tau}) \\ &= e(g, W)^{r\lambda_\tau} \cdot e(g, V)^{rt_i} \cdot e(g, V)^{-rt_i} \\ &= e(g, W)^{r\lambda_\tau}. \end{aligned}$$

In the case of $\pi(i) = x'_i$, we have

$$\begin{aligned} & e(C_{i,1}, D_2) \cdot \prod_{j \in [k]} \big(e(C_{i,3}, K'_{j,1}) \cdot e(C_{i,2}, K'_{j,2})\big)^{\frac{1}{x_i - \omega_j}} \\ = \; & e(W^{\lambda_i} \cdot U^{bt_i}, g^r) \cdot \prod_{j \in [k]} \Big(e\big((U^{x_i} H)^{-t_i}, g^{br'_j}\big) \cdot e\big(g^{t_i}, (U^{b\omega_j} H^b)^{r'_j}\big)\Big)^{\frac{1}{x_i - \omega_j}} \\ = \; & e(g, W)^{r\lambda_i} \cdot e(g, U)^{rbt_i} \cdot \prod_{j \in [k]} e(g, U)^{-r'_j bt_i} = e(g, W)^{r\lambda_i}. \end{aligned}$$

E Proof of Security of Our TIBBE Scheme in Section 6

In this section, we first introduce a new assumption that we call the n-(A) assumption. Then we show the assumption to hold in the generic bilinear group model. Finally, we show that our scheme in Sec. 6 is secure under the n-(A) assumption.

E.1 Definition of the n-(A) Assumption

n-(A) Assumption. Let $x, y, z, a_1, \ldots, a_n, b_1, \ldots, b_n \xleftarrow{\$} \mathbb{Z}_p$ and $g \xleftarrow{\$} \mathbb{G}^*$. We define Ψ as $\Psi =$

$$\begin{array}{l} g, g^x, g^y, g^z \\ g^{(xz)^2} \\ g^{a_i}, g^{xza_i}, g^{xz/a_i}, g^{x^2za_i}, g^{y/a_i^2}, g^{y^2/a_i^2} \quad \forall i \in [n] \\ g^{xza_i/a_j}, g^{ya_i/a_j^2}, g^{xyza_i/a_j^2}, g^{(xz)^2 a_i/a_j} \quad \forall (i,j) \in [n,n], i \neq j \\ g^{b_i}, g^{zb_i}, g^{b_i b_j}, g^{xy/b_i^2} \quad \forall (i,j) \in [n,n] \\ g^{zb_i b_j}, g^{xyb_j/b_i^2}, g^{xyb_i b_j/b_k^2}, g^{xyb_i^2/b_j^2} \quad \forall (i,j,k) \in [n,n,n], i \neq j. \end{array}$$

We say that an adversary $\mathcal{A}$ breaks n-(A) assumption on $(\mathbb{G}, \mathbb{G}_T)$ if $\mathcal{A}$ runs in polynomial time and $\frac{1}{2}|\Pr[\mathcal{A}(\Psi, e(g,g)^{xyz}) \to 0] - \Pr[\mathcal{A}(\Psi, T) \to 0]|$ is negligible where $T \xleftarrow{\$} \mathbb{G}_T$.

One can think that terms in Ψ comes from the q-MEDDH assumption in [21] and the q-1 ssumption in [30].

E.2 Generic Security of the n-(A) Assumption

Here, we briefly show that the assumption to hold in the generic group model. Actually, the assumption can be seen as an instance of $\mathbb{G}_T$-monomial assumption introduced in [30]. Thus, by the corollary D.4 in [30], it suffices to show that pairing result of any two elements from Ψ is not (symbolically) equivalent to $e(g,g)^{xyz}$. We can divide Ψ into theree parts as

$$\Psi_1 = \{g, g^x, g^y, g^z\},$$

$$\Psi_2 = \left\{ \begin{array}{ll} g^{(xz)^2} & \\ g^{a_i}, g^{xza_i}, g^{xz/a_i}, g^{x^2za_i}, g^{y/a_i^2}, g^{y^2/a_i^2} & \forall i \in [n] \\ g^{xza_i/a_j}, g^{ya_i/a_j^2}, g^{xyza_i/a_j^2}, g^{(xz)^2a_i/a_j} & \forall (i,j) \in [n,n], i \neq j \end{array} \right\},$$

$$\Psi_3 = \left\{ \begin{array}{ll} g^{b_i}, g^{zb_i}, g^{b_ib_j}, g^{xy/b_i^2} & \forall (i,j) \in [n,n] \\ g^{zb_ib_j}, g^{xyb_j/b_i^2}, g^{xyb_ib_j/b_k^2}, g^{xyb_i^2/b_j^2} & \forall (i,j,k) \in [n,n,n], i \neq j \end{array} \right\}.$$

One can think that the terms in Ψ_1 and Ψ_2 comes from the q-1 ssumption in [30] while Ψ_1 and Ψ_3 comes from the q-MEDDH assumption in [21]. We can show that the pairing computation of any two elements within $\Psi_1 \cup \Psi_2$ (resp., $\Psi_1 \cup \Psi_3$) does not give rise to the element $e(g,g)^{xyz}$ as in [30] (resp., [21]). It is easy to see that the paring computation of any two elements within $\Psi_2 \cup \Psi_3$ does not give rise to the element either, since a_i and b_j in the exponent never cancel out. Therefore, the assumption is secure in the generic group model.

E.3 Proof of Security for Our TIBBE Scheme in Section 6

Theorem 4. *Suppose the n-(A) assumption holds. Then, no PPT adversary can break the selective security of our scheme with non-negligible advantage with a challenge set of size $k \leq n$.*

Our proof proceeds similarly to that of the IBR scheme in [21] and that of KP-ABE scheme in [30].

Proof. We construct an algorithm $\mathcal{B}$ that receives problem instance of the n-(A) assumption and decides if $T = e(g,g)^{xyz}$ using the selective adversary $\mathcal{A}$ against our scheme.
Setup of master public key. At the outset of the game, the adversary $\mathcal{A}$ declares challenge set $S^\star = (\mathsf{ID}_1^\star, \ldots, \mathsf{ID}_k^\star)$ where $k \leq n$. $\mathcal{B}$ then picks $\tilde{u}, \tilde{h}, \tilde{v} \xleftarrow{\$} \mathbb{Z}_p$ and computes

$$g = g, \quad U = g^{\tilde{u}} \cdot \prod_{i \in [k]} g^{y/a_i^2}, \quad H = g^{\tilde{h}} \cdot \prod_{i \in [k]} g^{xz/a_i} \cdot \prod_{i \in [k]} (g^{y/a_i^2})^{-\mathsf{ID}_i^\star}, \quad e(g,g)^\alpha = e(g^x, g^y)$$
$$W = g^x, \quad B = \prod_{i \in [k]} g^{b_i}, \qquad V = g^{\tilde{v}} \cdot \prod_{i \in [k]} (g^{b_i})^{-\mathsf{ID}_i^\star}.$$

In the above, $\mathcal{B}$ implicitly sets $\alpha = xy$ and $b = \sum_{i \in [k]} b_i$. $\mathcal{B}$ also computes $B' = g^{b^2} = \prod_{(i,j) \in [k,k]} g^{b_ib_j}$ and

$$V' = V^b = \Big(g^{\tilde{v}} \cdot \prod_{i \in [k]} (g^{b_i})^{-\mathsf{ID}_i^\star} \Big)^{\sum_{j \in [k]} b_j} = \big(\prod_{j \in [k]} g^{b_j} \big)^{\tilde{v}} \cdot \prod_{(i,j) \in [k,k]} (g^{b_ib_j})^{-\mathsf{ID}_i^\star}.$$

Then $\mathcal{B}$ gives master public key $\mathsf{mpk} = (g, H, U, W, B, B', V, V', e(g,g)^\alpha)$ to $\mathcal{A}$.
Phase1 and 2. When $\mathcal{A}$ queries private key for $(\mathsf{ID}, \mathsf{type})$, $\mathcal{B}$ proceeds as follows. There are two cases to distinguish.

- In the case of $\mathsf{type} = \mathsf{IBBE}$ and $\mathsf{ID} \notin S^\star$, $\mathcal{B}$ chooses $\tilde{r} \xleftarrow{\$} \mathbb{Z}_p$ and implicitly sets r as

$$r = \tilde{r} - y + \sum_{i \in [k]} \frac{xza_i}{\mathsf{ID} - \mathsf{ID}_i^\star}.$$

  We remark that r is well-defined because $\mathsf{ID} \notin S^\star = \{\mathsf{ID}_1^\star \ldots, \mathsf{ID}_k^\star\}$ and thus the denominators $\{\mathsf{ID} - \mathsf{ID}_i^\star\}_{i \in [k]}$ are non zero. Furthermore, r is properly distributed due to $\tilde{r}$. $\mathcal{B}$ then computes

the private key $\mathsf{sk}_{(\mathsf{ID},\mathsf{IBBE})} = \Big(D_1 = g^\alpha \cdot W^r, D_2 = (U^{\mathsf{ID}}H)^r, D_3 = g^r\Big)$ as follows:

$$
\begin{aligned}
D_1 &= g^\alpha \cdot W^r = g^{xy} \cdot (g^x)^{-y+\sum_{i\in[k]} xza_i/(\mathsf{ID}-\mathsf{ID}_i^\star)+\tilde{r}} = \prod_{i\in[k]} (g^{x^2za_i})^{1/(\mathsf{ID}-\mathsf{ID}_i^\star)} \cdot (g^x)^{\tilde{r}} \\
D_2 &= (U^{\mathsf{ID}}H)^r \\
&= (U^{\mathsf{ID}}H)^{\tilde{r}} \cdot \Big(g^{\tilde{u}\mathsf{ID}+\tilde{h}} \cdot \prod_{i\in[k]} g^{xz/a_i} \cdot \prod_{i\in[k]} g^{(\mathsf{ID}-\mathsf{ID}_i^\star)y/a_i^2}\Big)^{-y+\sum_{j\in[k]} xza_j/(\mathsf{ID}-\mathsf{ID}_j^\star)} \\
&= \underbrace{(U^{\mathsf{ID}}H)^{\tilde{r}} \cdot \Big((g^y)^{-1} \cdot \prod_{j\in[k]} (g^{xza_j})^{1/\mathsf{ID}-\mathsf{ID}_j^\star}\Big)^{\tilde{u}\mathsf{ID}+\tilde{h}}}_{\Phi_1} \cdot \prod_{i\in[k]} g^{-xyz/a_i} \\
&\quad \underbrace{\cdot \prod_{i\in[k]} (g^{y^2/a_i^2})^{\mathsf{ID}_i^\star-\mathsf{ID}} \cdot \prod_{(i,j)\in[k,k]} (g^{(xz)^2a_j/a_i})^{1/(\mathsf{ID}-\mathsf{ID}_j^\star)}}_{\Phi_2} \cdot \prod_{(i,j)\in[k,k]} (g^{xyza_j/a_i^2})^{(\mathsf{ID}-\mathsf{ID}_i^\star)/(\mathsf{ID}-\mathsf{ID}_j^\star)} \\
&= \Phi_1 \cdot \Phi_2 \cdot \prod_{\substack{(i,j)\in[k,k] \\ i\neq j}} (g^{xyza_j/a_i^2})^{(\mathsf{ID}-\mathsf{ID}_i^\star)/(\mathsf{ID}-\mathsf{ID}_j^\star)} \\
D_3 &= g^r = (g^y)^{-1} \cdot \prod_{i\in[k]} (g^{xza_i})^{1/(\mathsf{ID}-\mathsf{ID}_i^\star)} \cdot g^{\tilde{r}}.
\end{aligned}
$$

Finally, $\mathcal{B}$ gives $\mathsf{sk}_{\mathsf{ID},\mathsf{IBBE}} = (D_1, D_2, D_3)$ to $\mathcal{A}$.

- In the case of $\mathsf{type} = \mathsf{IBR}$ and $\mathsf{ID} \in S^\star$, we have $\mathsf{ID} = \mathsf{ID}_\tau^\star$ for some $\tau \in [k]$. In this case, $\mathcal{B}$ chooses $\tilde{r} \xleftarrow{\$} \mathbb{Z}_p$ and implicitly sets r as

$$r = \tilde{r} - xy/b_\tau^2.$$

We have that r is properly distributed due to $\tilde{r}$. $\mathcal{B}$ then computes the private key $\mathsf{sk}_{(\mathsf{ID},\mathsf{IBR})} = \Big(D_1 = g^\alpha \cdot B'^r, D_2 = (B^{\mathsf{ID}}V)^r, D_3 = g^r\Big)$ as follows:

$$
\begin{aligned}
D_1 &= g^\alpha \cdot (B')^r = g^{xy} \cdot \Big(\prod_{(i,j)\in[k,k]} g^{b_ib_j}\Big)^{-xy/b_\tau^2} \cdot (B')^{\tilde{r}} \\
&= \prod_{\substack{(i,j)\in[k,k] \\ (i,j)\neq(\tau,\tau)}} g^{-xyb_ib_j/b_\tau^2} \cdot (B')^{\tilde{r}} = \prod_{\substack{(i,j)\in[k,k] \\ i\neq j}} (g^{xyb_ib_j/b_\tau^2})^{-1} \cdot \prod_{i\in[k]\setminus\{\tau\}} (g^{xyb_i^2/b_\tau^2})^{-1} \cdot (B')^{\tilde{r}} \\
D_2 &= (B^{\mathsf{ID}}V)^r = (B^{\mathsf{ID}}V)^{\tilde{r}} \cdot \Big(g^{\tilde{v}} \cdot \prod_{i\in[k]} (g^{b_i})^{\mathsf{ID}_\tau^\star-\mathsf{ID}_i^\star}\Big)^{-xy/b_\tau^2} \\
&= (B^{\mathsf{ID}}V)^{\tilde{r}} \cdot (g^{xy/b_\tau^2})^{-\tilde{v}} \cdot \prod_{i\in[k]\setminus\{\tau\}} (g^{xyb_i/b_\tau^2})^{\mathsf{ID}_i^\star-\mathsf{ID}_\tau^\star} \\
D_3 &= g^{\tilde{r}} \cdot (g^{xy/b_\tau^2})^{-1}.
\end{aligned}
$$

Finally, $\mathcal{B}$ gives $\mathsf{sk}_{(\mathsf{ID},\mathsf{IBR})} = (D_1, D_2, D_3)$ to $\mathcal{A}$.

Challenge. At some point in the game, $\mathcal{A}$ submits a pair of ciphertexts $(\mathsf{M}_0, \mathsf{M}_1)$ to $\mathcal{B}$. $\mathcal{B}$ flips a random coin $\beta \xleftarrow{\$} \{0,1\}$ and sets $(C_0, C_1) = (\mathsf{M}_\beta \cdot T, g^z)$ where T is the challenge term and g^z is

the corresponding term of the assumption. Additionally, it has to compute $\{C_{i,1}, C_{i,2}, C'_{i,1}, C'_{i,2}\}$ for $i \in [k]$.

- $\mathcal{B}$ chooses $\tilde{t}_\tau \xleftarrow{\$} \mathbb{Z}_p$ and implicitly sets $t_\tau = \tilde{t}_\tau - a_\tau$. Then $\mathcal{B}$ can compute $C_{\tau,1}$ and $C_{\tau,2}$ for all $\tau \in [k]$ as follows.

$$\begin{aligned}
C_{\tau,1} &= W^{-s}(U^{\mathsf{ID}^\star_\tau} H)^{-t_\tau} \\
&= g^{-xz} \cdot \Big(g^{\tilde{u}\mathsf{ID}^\star_\tau + \tilde{h}} \cdot \prod_{i\in[k]} g^{xz/a_i} \cdot \prod_{i\in[k]} (g^{y/a_i^2})^{\mathsf{ID}^\star_\tau - \mathsf{ID}^\star_i}\Big)^{a_\tau} \cdot (U^{\mathsf{ID}^\star_\tau} H)^{-\tilde{t}_\tau} \\
&= g^{-xz} \cdot (g^{a_\tau})^{\tilde{u}\mathsf{ID}^\star_\tau + \tilde{h}} \cdot \prod_{i\in[k]} g^{xza_\tau/a_i} \cdot \prod_{i\in[k]} (g^{ya_\tau/a_i^2})^{\mathsf{ID}^\star_\tau - \mathsf{ID}^\star_i} \cdot (U^{\mathsf{ID}^\star_\tau} H)^{-\tilde{t}_\tau} \\
&= (g^{a_\tau})^{\tilde{u}\mathsf{ID}^\star_\tau + \tilde{h}} \cdot \prod_{i\in[k]\setminus\{\tau\}} g^{xza_\tau/a_i} \cdot \prod_{i\in[k]\setminus\{\tau\}} (g^{ya_\tau/a_i^2})^{\mathsf{ID}^\star_\tau - \mathsf{ID}^\star_i} \cdot (U^{\mathsf{ID}^\star_i} H)^{-\tilde{t}_\tau} \\
C_{\tau,2} &= g^{\tilde{t}_\tau} \cdot (g^{a_\tau})^{-1}.
\end{aligned}$$

- $\mathcal{B}$ chooses random $\tilde{s}_1, \ldots, \tilde{s}_k$ such that $\sum_{i\in[k]} \tilde{s}_i = 0$ and implicitly sets s_i as $s_i = \tilde{s}_i + b_i z/b$ for all $i \in [k]$. We have $\sum_{i\in[k]} s_i = \sum_{i\in[k]} (\tilde{s}_i + b_i z/b) = 0 + bz/b = z$ and $\{s_i\}_{i\in[k]}$ are properly distributed. $\mathcal{B}$ can compute $C'_{\tau,1}$ and $C'_{\tau,2}$ for all $\tau \in [k]$ as follows.

$$\begin{aligned}
C'_{\tau,1} &= \big(B'^{\mathsf{ID}^\star_\tau} V'\big)^{-s_\tau} = \big(g^{b^2\mathsf{ID}^\star_\tau} V^b\big)^{-zb_\tau/b} \cdot \big(B'^{\mathsf{ID}^\star_\tau} V'\big)^{-\tilde{s}_\tau} \\
&= \big(g^{\tilde{v}} \cdot \prod_{i\in[k]} (g^{b_i})^{\mathsf{ID}^\star_\tau - \mathsf{ID}^\star_i}\big)^{-zb_\tau} \cdot \big(B'^{\mathsf{ID}^\star_\tau} V'\big)^{-\tilde{s}_\tau} \\
&= (g^{zb_\tau})^{-\tilde{v}} \cdot \Big(\prod_{i\in[k]\setminus\{\tau\}} (g^{zb_ib_\tau})^{\mathsf{ID}^\star_i - \mathsf{ID}^\star_\tau}\Big) \cdot \big(B'^{\mathsf{ID}^\star_\tau} V'\big)^{-\tilde{s}_\tau} \\
C'_{\tau,2} &= B^{s_\tau} = B^{\tilde{s}_\tau} \cdot g^{zb_\tau}.
\end{aligned}$$

After all the above steps, $\mathcal{B}$ gives the challenge ciphertext $(C_0, C_1, \{C_{i,1}, C_{i,2}, C'_{i,1}, C'_{i,2}\}_{i\in[k]})$ to $\mathcal{A}$.

Guess. Finally, $\mathcal{A}$ outputs its guess β' for β. If $\beta' = \beta$, $\mathcal{A}$ outputs 1 for its guess. Otherwise, it outputs 0. If $T = e(g,g)^{xyz}$, the above simulation is perfect and thus $\mathcal{A}$ has non-negligible advantage. On the other hand, If T is a random element in $\mathbb{G}_T$, $\mathcal{A}$'s advantage is 0. Therefore, if $\mathcal{A}$ breaks our scheme with non-negligible advantage, $\mathcal{B}$ has a non-negligible advantage in breaking the n-(A) assumption. □

F Proof of Security for Our CP-ABE Scheme in Section 7

In this section, we first introduce a new assumption that we call the n-(B) assumption. Then we show the assumption to hold in the generic bilinear group model. Finally, we show that our scheme in Sec. 7 is secure under the n-(B) assumption.

F.1 Definition of the n-(B) Assumption

n-(B) Assumption. Let $s, a, b_1 \ldots, b_n \xleftarrow{\$} \mathbb{Z}_p$ and $g \xleftarrow{\$} \mathbb{G}^*$. We define Ψ as $\Psi =$

$$\begin{aligned}
&g, g^s \\
&g^{a^i}, g^{b_j}, g^{sb_j}, g^{sb_ib_j}, g^{a^ib_j}, g^{a^i/b_j^2} \quad \forall (i,j) \in [n,n]
\end{aligned}$$

$$
\begin{aligned}
&g^{a^i/b_j} \quad \forall (i,j) \in [2n, n], i \neq n+1 \\
&g^{a^i b_j / b_{j'}^2} \quad \forall (i,j,j') \in [2n, n, n], j \neq j' \\
&g^{sa^i b_j / b_{j'}}, g^{sa^i b_j / b_{j'}^2} \quad \forall (i,j,j') \in [n, n, n], j \neq j' \\
&g^{sa^i b_j b_{j'} / b_{j''}^2} \quad \forall (i,j,j',j'') \in [n, n, n, n], j \neq j'', j' \neq j''
\end{aligned}
$$

We say that an adversary $\mathcal{A}$ breaks n-(B) assumption on $(\mathbb{G}, \mathbb{G}_T)$ if $\mathcal{A}$ runs in polynomial time and $\frac{1}{2}|\Pr[\mathcal{A}(\Psi, e(g,g)^{sa^{n+1}}) \to 0] - \Pr[\mathcal{A}(\Psi, T) \to 0]|$ is negligible where $T \xleftarrow{\$} \mathbb{G}_T$.

The assumption is very similar to q-1 assumption introduced in [30]. The difference is that in the n-(B) assumption above, we have some additional terms in the problem instance.

F.2 Generic Security of the n-(B) Assumption

Here, we briefly show that the assumption to hold in the generic group model. Actually, the assumption can be seen as an instance of $\mathbb{G}_T$-monomial assumption introduced in [30]. Thus, by the corollary D.4 in [30], it suffices to show that pairing result of any two elements from Φ is not (symbolically) equivalent to $e(g,g)^{a^{n+1}s}$. Note that our assumption is the same as q-1 assumption in [30], which is proven secure in the generic group model, except that there is additional terms $g^{sb_ib_j}$ and $g^{sa^ib_jb_{j'}/b_{j''}^2}$ in the problem instance. These additional terms does not harm the security of the assumption since there is no term as g^{a^{n+1}/b_ib_j} nor $g^{a^ib_{j''}^2/b_jb_{j'}}$ for any i, j, j', j'' such that $j \neq j'' \wedge j' \neq j''$ in the problem instace. Therefore, n-(B) assumption is secure in the generic group model.

F.3 Proof of Security for Our CP-ABE Scheme in Section 7

Theorem 5. *Suppose the n-(B) assumption holds. Then, no PPT adversary can break the selective security of our CP-ABE scheme in Sec. 7 with non-negligible advantage with a challenge matrix of size $\ell \times m$ where $\ell, m \leq n$.*

Since our scheme is an extension of the CP-ABE scheme proposed in [30], the basic strategy for the security proof is similar. The difference is that we have to simulate components related to negation of attribute. This can be done by extending the proof technique in [21] to our setting.

Proof. We construct an algorithm $\mathcal{B}$ that receives problem instance of the n-(B) assumption and decides if $T = e(g,g)^{a^{n+1}s}$ using the selective adversary $\mathcal{A}$ against our scheme.

Setup of master public key. At the outset of the game, the adversary $\mathcal{A}$ declares challenge policy $\tilde{\mathbb{A}}^\star$ where $\tilde{\mathbb{A}}^\star = NM(\mathbb{A}^\star)$ and $\mathbb{A}^\star$ is specified by $(L^\star, \pi^\star)$. We have that $L^\star$ is an $\ell \times m$ matrix, where $\ell, m \leq n$. We divide $[\ell]$ into two sets. We define $\mathsf{posi} = \{i | i \in [\ell] \wedge \pi^\star(i) = x_i\}$ and $\mathsf{nega} = \{i | i \in [\ell] \wedge \pi^\star(i) = x_i'\}$. That is, posi (resp., nega) is a set of indices that is associated with non-negated (resp., negated) attribute by $\pi^\star$.

$\mathcal{B}$ then picks $\tilde{\alpha}, \tilde{u}, \tilde{v}, \tilde{h} \xleftarrow{\$} \mathbb{Z}_p$ and computes

$$
\begin{aligned}
&g = g, \quad H = g^{\tilde{h}} \cdot \prod_{(j,k) \in [\ell,m]} \left(g^{a^k/b_j^2}\right)^{-\pi^\star(j) L^\star_{j,k}}, \quad U = g^{\tilde{u}} \cdot \prod_{(j,k) \in [\ell,m]} \left(g^{a^k/b_j^2}\right)^{L^\star_{j,k}} \\
&W = g^a, \quad V = g^{\tilde{v}} \cdot \prod_{(j,k) \in \mathsf{posi} \times [m]} \left(g^{a^k/b_j}\right)^{L^\star_{j,k}}, \qquad e(g,g)^\alpha = e(g,g)^{\tilde{\alpha}} \cdot e(g^a, g^{a^n}).
\end{aligned}
$$

We remark that in the computation of H above, we treat primed (i.e., negated) attributes in exponents as elements in $\mathbb{Z}_p$. That is, for a group element $A \in \mathbb{G}$ and a primed attribute x_i', we defne $A^{x_i'} := A^{x_i}$.

Here, $\mathcal{B}$ implicitly sets $\alpha = \tilde{\alpha} + a^{n+1}$. $\mathcal{B}$ also implicitly sets $b = \sum_{i \in \mathsf{nega}} b_i$ and computes

$$\begin{aligned} V' &= \Big(g^{\tilde{u}} \cdot \prod_{(j,k)\in[\ell,m]} (g^{a^k/b_j^2})^{L^\star_{j,k}}\Big)^{\sum_{i\in\mathsf{nega}} b_i} = \Big(\prod_{i\in\mathsf{nega}} g^{b_i}\Big)^{\tilde{u}} \cdot \prod_{(i,j,k)\in\mathsf{nega}\times[\ell,m]} \big(g^{a^k b_i/b_j^2}\big)^{L^\star_{j,k}} \\ &= \Big(\prod_{i\in\mathsf{nega}} g^{b_i}\Big)^{\tilde{u}} \cdot \prod_{\substack{(i,j,k)\in\mathsf{nega}\times[\ell,m]\\ i\neq j}} \big(g^{a^k b_i/b_j^2}\big)^{L^\star_{j,k}} \cdot \prod_{(j,k)\in\mathsf{nega}\times[m]} \big(g^{a^k/b_j}\big)^{L^\star_{j,k}}. \end{aligned}$$

Then $\mathcal{B}$ gives $\mathsf{mpk} = (g, H, U, V, V', W, e(g,g)^\alpha)$ to $\mathcal{A}$. One can easily verify that mpk computed as above is properly distributed.

Phase1 and 2. When $\mathcal{A}$ queries private key for an attribute set $\omega = \{\omega_1, \ldots, \omega_{|\omega|}\}$, $\mathcal{B}$ answers as follows. In both phases, the treatment is the same.

Since $\omega \notin \tilde{\mathbb{A}}^\star$, we have that $\omega' = N(\omega) \notin \mathbb{A}^\star$. Therefore $\mathbf{1} = (1, 0, \ldots, 0)$ does not lie in the row space of $L^\star_{\omega'}$, which is the submatrix of $L^\star$ formed by rows corresponding to attributes in ω'. Hence, due to the proposition 11 in [16], we have that there must exist an efficiently computable vector $\mathbf{z} = (z_1, \ldots, z_m) \in \mathbb{Z}_p^m$ such that $\langle \mathbf{1}, \mathbf{z}\rangle = 1$ and $L_{\omega'} \cdot \mathbf{z}^\top = \mathbf{0}$. $\mathcal{B}$ chooses $\tilde{r} \xleftarrow{\$} \mathbb{Z}_p$ and implicitly sets

$$r = \tilde{r} - (z_1 a^n + z_2 a^{n-1} + \cdots + z_m a^{n+1-m}) = \tilde{r} - \sum_{i\in[m]} z_i a^{n+1-i}.$$

This is properly distributed due to $\tilde{r}$. Then using the suitable terms from the assumption it can computes

$$\begin{aligned} D_1 &= g^\alpha W^r = g^{a^{n+1}} g^{\tilde{\alpha}} g^{a\tilde{r}} \prod_{i\in[m]} g^{-z_i a^{n+2-i}} = g^{\tilde{\alpha}} (g^a)^{\tilde{r}} \prod_{i=2}^{m} \big(g^{a^{n+2-i}}\big)^{-z_i} \\ D_2 &= g^r = g^{\tilde{r}} \prod_{i\in[m]} \big(g^{a^{n+1-i}}\big)^{-z_i}. \end{aligned}$$

Additionally, it has to compute $\{K_{i,1}, K_{i,2}, K'_{i,1}, K'_{i,2}\}$ for $i \in [|\omega|]$.

- At first, we explain how to compute $K_{\tau,1} = V^{-r}(U^{\omega_\tau} H)^{r_\tau}, K_{\tau,2} = g^{r_\tau}$ for $\tau \in [|\omega|]$. The common part V^{-r} is as follows:

$$\begin{aligned} V^{-r} &= V^{-\tilde{r}} \cdot \big(g^{\tilde{v}} \cdot \prod_{(j,k)\in\mathsf{posi}\times[m]} g^{L^\star_{j,k} a^k/b_j}\big)^{\sum_{i\in[m]} z_i a^{n+1-i}} \\ &= V^{-\tilde{r}} \cdot \prod_{i\in[m]} \big(g^{a^{n+1-i}}\big)^{\tilde{v} z_i} \cdot \prod_{(i,j,k)\in[m]\times\mathsf{posi}\times[m]} g^{z_i L^\star_{j,k} a^{n+1+k-i}/b_j} \\ &= \underbrace{V^{-\tilde{r}} \cdot \prod_{i\in[m]} \big(g^{a^{n+1-i}}\big)^{\tilde{v} z_i} \cdot \prod_{\substack{(i,j,k)\in[m]\times\mathsf{posi}\times[m]\\ i\neq k}} \big(g^{a^{n+1+k-i}/b_j}\big)^{z_i L^\star_{j,k}}}_{\Phi} \cdot \prod_{(i,j)\in[m]\times\mathsf{posi}} g^{z_i L^\star_{j,i} a^{n+1}/b_j} \\ &= \Phi \cdot \prod_{j\in\mathsf{posi}} g^{\langle \mathbf{L}^\star_j, \mathbf{z}\rangle a^{n+1}/b_j} \\ &= \Phi \cdot \prod_{\substack{j\in\mathsf{posi}\\ \pi(j)\notin\omega}} \big(g^{a^{n+1}/b_j}\big)^{\langle \mathbf{L}^\star_j, \mathbf{z}\rangle} \end{aligned}$$

The Φ part can be computed by the simulator using appropriate term, while the second term has to be canceled by the $(U^{\omega_\tau}H)^{r_\tau}$ part. So for $\tau \in [|\omega|]$, the simulator sets implicitly

$$r_\tau = \tilde{r}_\tau - \sum_{\substack{(i,i')\in[m]\times\mathsf{posi}\\ \pi^\star(i')\notin\omega}} \frac{z_i b_{i'} a^{n+1-i}}{\omega_\tau - \pi^\star(i')}$$

where $\tilde{r}_\tau \xleftarrow{\$} \mathbb{Z}_p$ and therefore r_τ is properly distributed. We remark that r_τ is well-defined because $\omega_\tau \neq \pi^\star(i')$ for all i' such that $(i' \in \mathsf{posi}) \wedge (\pi^\star(i') \notin \omega)$ and thus the denominators $\omega_\tau - \pi^\star(i')$ are non zero. The second part of $V^{-r}(U^{\omega_\tau}H)^{r_\tau}$ are as follows:

$$\begin{aligned}
(U^{\omega_\tau}H)^{r_\tau} &= (U^{\omega_\tau}H)^{\tilde{r}_\tau} \cdot \Big(g^{\tilde{u}\omega_\tau+\tilde{h}} \cdot \prod_{(j,k)\in[\ell,m]} g^{(\omega_\tau-\pi^\star(j))L^\star_{j,k}a^k/b_j^2}\Big)^{-\sum_{\substack{(i,i')\in[m]\times\mathsf{posi}\\ \pi^\star(i')\notin\omega}} \frac{z_i b_{i'} a^{n+1-i}}{\omega_\tau-\pi^\star(i')}} \\
&= \underbrace{(U^{\omega_\tau}H)^{\tilde{r}_\tau} \cdot \Big(\prod_{\substack{(i,i')\in[m]\times\mathsf{posi}\\ \pi^\star(i')\notin\omega}} (g^{b_{i'}a^{n+1-i}})^{-z_i/(\omega_\tau-\pi^\star(i'))}\Big)^{\tilde{u}\omega_\tau+\tilde{h}}}_{\Phi_1} \\
&\quad \cdot \prod_{\substack{(i',i,j,k)\in\mathsf{posi}\times[m,\ell,m]\\ \pi^\star(i')\notin\omega}} g^{-(\omega_\tau-\pi^\star(j))L^\star_{j,k}z_i b_{i'}a^{n+1+k-i}/(\omega_\tau-\pi^\star(i'))b_j^2} \\
&= \Phi_1 \cdot \underbrace{\prod_{\substack{(i',i,j,k)\in\mathsf{posi}\times[m,\ell,m]\\ \pi^\star(i')\notin\omega,(j\neq i')\vee(i\neq k)}} \big(g^{b_{i'}a^{n+1+k-i}/b_j^2}\big)^{-(\omega_\tau-\pi^\star(j))L^\star_{j,k}z_i/(\omega_\tau-\pi^\star(i'))}}_{\Phi_2} \\
&\quad \cdot \prod_{\substack{(i',i,j,k)\in\mathsf{posi}\times[m,\ell,m]\\ \pi^\star(i')\notin\omega,(j=i')\wedge(i=k)}} \big(g^{b_{i'}a^{n+1+k-i}/b_j^2}\big)^{-(\omega_\tau-\pi^\star(j))L^\star_{j,k}z_i/(\omega_\tau-\pi^\star(i'))} \\
&= \Phi_1\cdot\Phi_2 \cdot \prod_{\substack{(j,k)\in\mathsf{posi}\times[m]\\ \pi^\star(j)\notin\omega}} (g^{a^{n+1}/b_j})^{-L^\star_{j,k}z_k} \\
&= \Phi_1\cdot\Phi_2 \cdot \prod_{\substack{j\in\mathsf{posi}\\ \pi^\star(j)\notin\omega}} (g^{a^{n+1}/b_j})^{-\langle \mathbf{L}^\star_j,\mathbf{z}\rangle}
\end{aligned}$$

One can verify that Φ_1 and Φ_2 are efficiently computable using appropriate terms given to $\mathcal{B}$. Since the problematic part of V^{-r} cancels out with $(U^{\omega_\tau}H)^{r_\tau}$, $\mathcal{B}$ can compute $K_{\tau,1} = \Phi\cdot\Phi_1\cdot\Phi_2$ for $\tau \in [|\omega|]$ efficiently. $\mathcal{B}$ can also effciently compute $K_{\tau,2}$ as follows.

$$g^{r_\tau} = g^{\tilde{r}_\tau} \cdot \prod_{\substack{(i,i')\in[m]\times\mathsf{posi}\\ \pi^\star(i')\notin\omega}} (g^{b_{i'}a^{n+1-i}})^{-z_i/(\omega_\tau-\pi^\star(i'))}$$

- Next, we explain how to compute $K'_{i,1} = (U^{b\omega_i}H^b)^{r'_i}, K'_{i,2} = g^{br'_i}$ for $i \in [|\omega|]$ (with the constraint that $\sum_{i\in[|\omega|]} r'_i = r$). These terms are computed step-by-step. At first, $\mathcal{B}$ sets $K'_{i,1} = 1_{\mathbb{G}}, K'_{i,2} = 1_{\mathbb{G}}$ for $i \in [|\omega|]$ and updates these values as follows. For all $\tau \in \mathsf{nega}$, $\mathcal{B}$ executes the following steps. There are two cases to distinguish:

– In the case that there is $\nu \in [|\omega|]$ such that $\pi^\star(\tau) = x'_\tau \wedge x_\tau = \omega_\nu \in \omega$ (i.e., $\pi^\star(\tau) \notin N(\omega)$), $\mathcal{B}$ updates $K'_{\nu,1}$ and $K'_{\nu,2}$ as

$$K'_{\nu,1} \leftarrow K'_{\nu,1} \cdot (U^{\omega_\nu} H)^{b_\tau r}, \quad K'_{\nu,2} \leftarrow K'_{\nu,2} \cdot g^{b_\tau r}.$$

In the following, we show that $(U^{\omega_\nu} H)^{b_\tau r}$ and $g^{b_\tau r}$ can be computed efficiently. First, we can see that

$$g^{b_\tau r} = (g^{b_\tau})^{\tilde{r}} \cdot \prod_{i \in [m]} (g^{b_\tau a^{n+1-i}})^{-z_i}$$

can be computed efficiently. Next, we see that

$$\begin{aligned}
(U^{\omega_\nu} H)^{b_\tau r} &= (U^{\pi^\star(\tau)} H)^{b_\tau r} \\
&= (g^{b_\tau r})^{\pi^\star(\tau)\tilde{u}+\tilde{h}} \cdot \Big(\prod_{(j,k) \in [\ell,m]} g^{(\pi^\star(\tau)-\pi^\star(j)) L^\star_{j,k} a^k / b_j^2} \Big)^{b_\tau (\tilde{r} - \sum_{i \in [m]} z_i a^{n+1-i})} \\
&= \underbrace{(g^{b_\tau r})^{\pi^\star(\tau)\tilde{u}+\tilde{h}} \cdot \prod_{(j,k) \in [\ell,m]} (g^{a^k b_\tau / b_j^2})^{\tilde{r}(\pi^\star(\tau)-\pi^\star(j)) L^\star_{j,k}}}_{\Phi} \\
&\qquad \cdot \prod_{(i,j,k) \in [m,\ell,m]} (g^{a^{n+1+k-i} b_\tau / b_j^2})^{-(\pi^\star(\tau)-\pi^\star(j)) L^\star_{j,k} z_i} \\
&= \Phi \cdot \prod_{\substack{(i,j,k) \in [m,\ell,m] \\ j \neq \tau}} (g^{a^{n+1+k-i} b_\tau / b_j^2})^{-(\pi^\star(\tau)-\pi^\star(j)) L^\star_{j,k} z_i}.
\end{aligned}$$

In the last equation above, the terms $g^{a^{n+1+k-i} b_\tau / b_\tau^2} = g^{a^{n+1+k-i}/b_\tau}$ for $(i,k) \in [m,m]$ (in particular, the problematic term g^{a^{n+1}/b_τ}) disappears. Thus, $\mathcal{B}$ can compute $(U^{\omega_\nu} H)^{b_\tau r}$ efficiently.

– In the case that $\pi^\star(\tau) = x'_\tau \wedge x_\tau \notin \omega$ (i.e., $\pi^\star(\tau) \in N(\omega)$), $\mathcal{B}$ updates $K'_{1,1}$ and $K'_{1,2}$[§] as

$$K'_{1,1} \leftarrow K'_{1,1} \cdot (U^{\omega_1} H)^{b_\tau r}, \quad K'_{1,2} \leftarrow K'_{1,2} \cdot g^{b_\tau r}.$$

In the following, we show that $(U^{\omega_1} H)^{b_\tau r}$ and $g^{b_\tau r}$ can be computed efficiently. As previous case, $g^{b_\tau r}$ can be computed efficiently using appropriate terms. Next, we see that

$$\begin{aligned}
(U^{\omega_1} H)^{b_\tau r} &= (g^{b_\tau r})^{\omega_1 \tilde{u}+\tilde{h}} \cdot \Big(\prod_{(j,k) \in [\ell,m]} g^{(\omega_1-\pi^\star(j)) L^\star_{j,k} a^k / b_j^2} \Big)^{b_\tau (\tilde{r} - \sum_{i \in [m]} z_i a^{n+1-i})} \\
&= \underbrace{(g^{b_\tau r})^{\omega_1 \tilde{u}+\tilde{h}} \cdot \prod_{(j,k) \in [\ell,m]} (g^{a^k b_\tau / b_j^2})^{\tilde{r}(\omega_1-\pi^\star(j)) L^\star_{j,k}}}_{\Phi_1} \\
&\qquad \cdot \prod_{(i,j,k) \in [m,\ell,m]} (g^{a^{n+1+k-i} b_\tau / b_j^2})^{-(\omega_1-\pi^\star(j)) L^\star_{j,k} z_i} \\
&= \Phi_1 \cdot \underbrace{\prod_{\substack{(i,j,k) \in [m,\ell,m] \\ (k \neq i) \vee (j \neq \tau)}} (g^{a^{n+1+k-i} b_\tau / b_j^2})^{-(\omega_1-\pi^\star(j)) L^\star_{j,k} z_i}}_{\Phi_2}
\end{aligned}$$

[§]Here, the choice $K'_{1,1}, K'_{1,2}$ is rather arbitrary. One can update any $(K'_{i,1}, K'_{i,2})$ as $K'_{i,1} \leftarrow K'_{i,1} \cdot (U^{\omega_i} H)^{b_\tau r}, K'_{i,2} \leftarrow K'_{i,2} \cdot g^{b_\tau r}$ instead.

$$
\begin{aligned}
&\cdot \prod_{\substack{(i,j,k)\in[m,\ell,m]\\(k=i)\wedge(j=\tau)}} (g^{a^{n+1+k-i}b_\tau/b_j^2})^{-(\omega_1-\pi^\star(j))L^\star_{j,k}z_i}\\
=\ & \Phi_1\cdot\Phi_2\cdot\prod_{i\in[m]}(g^{a^{n+1}/b_\tau})^{-(\omega_1-\pi^\star(\tau))L^\star_{\tau,i}z_i}\\
=\ & \Phi_1\cdot\Phi_2\cdot(g^{a^{n+1}/b_\tau})^{-(\omega_1-\pi^\star(\tau))\langle\mathbf{L}^\star_\tau,\mathbf{z}\rangle}\\
=\ & \Phi_1\cdot\Phi_2
\end{aligned}
$$

Since $\pi^\star(\tau)\in N(\omega)$, we have $\langle\mathbf{L}^\star_\tau,\mathbf{z}\rangle=0$ from the property of $\mathbf{z}$ and thus the problematic term g^{a^{n+1}/b_τ} disappers in the last equality above. One can verify that Φ_1 and Φ_2 are efficiently computable using appropriate terms.

After the above step, we have that $(K'_{i,1},K'_{i,2})$ for $i\in[|\omega|]$ is in the form of $(K'_{i,1},K'_{i,2})=((U^{\omega_i}H)^{\bar r'_i},g^{\bar r'_i})$ for some $\bar r'_i\in\mathbb{Z}_p$ such that $\sum_{i=1}^{|\omega|}\bar r'_i=\sum_{\tau\in\mathsf{nega}}b_\tau r=br$. By setting $\tilde r'_i=\bar r'_i/b$, we have

$$(K'_{i,1},K'_{i,2})=((U^{b\omega_i}H^b)^{\tilde r'_i},(g^b)^{\tilde r'_i})$$

and $\sum_{i=1}^{|\omega|}\tilde r'_i=(\sum_{i=1}^{|\omega|}\bar r'_i)/b=r$. Thus, $\{K'_{i,1},K'_{i,2}\}_{i=1}^{|\omega|}$ can be seen as a part of private key generated using randomness $\{\tilde r'_i\}_{i=1}^{|\omega|}$. However, $\{\tilde r'_i\}_{i=1}^{|\omega|}$ depend on $\{b_i\}_{i\in\mathsf{nega}}$ and thus $\mathcal{B}$ needs to rerandomize $\{K'_{i,1},K'_{i,2}\}_{i=1}^{|\omega|}$ so that it is independent from $\{b_i\}_{i\in\mathsf{nega}}$. To accomplish this, $\mathcal{B}$ chooses random $\hat r'_1,\ldots,\hat r'_{|\omega|}$ such that $\hat r'_1+\cdots+\hat r'_{|\omega|}=0$ and updates $\{K'_{i,1},K'_{i,2}\}_{i=1}^{|\omega|}$ as

$$K'_{i,1}\leftarrow K'_{i,1}\cdot(U^{\omega_i}H)^{\hat r'_i},\quad K'_{i,2}\leftarrow K'_{i,2}\cdot g^{\hat r'_i}.$$

Here, $\mathcal{B}$ implicitly sets $r'_i=\tilde r'_i+\hat r'_i/b$ for $i\in[|\omega|]$.

After all the above steps, $\mathcal{B}$ gives private key $\mathsf{sk}_\omega=\big(D_1,D_2,\{K_{i,1},K_{i,2},K'_{i,1},K'_{i,2}\}_{i\in[|\omega|]}\big)$ to $\mathcal{A}$.

Challenge. At some point in the game, $\mathcal{A}$ submits a pair of ciphertexts $(\mathsf{M}_0,\mathsf{M}_1)$ to $\mathcal{B}$. $\mathcal{B}$ flips a random coin $\beta\xleftarrow{\$}\{0,1\}$ and sets $(C_0,C_1)=(\mathsf{M}_\beta\cdot e(g^s,g^{\tilde\alpha})\cdot T,g^s)$ where T is the challenge term and g^s is the corresponding term of the assumption.

$\mathcal{B}$ chooses $\tilde s_2\ldots,\tilde s_m\xleftarrow{\$}\mathbb{Z}_p$ and implicitly sets $\mathbf{s}=(s,sa+\tilde s_2,sa^2+\tilde s_3,\ldots,sa^{m-1}+\tilde s_m)$. $\mathbf{s}$ is properly distributed due to $\{\tilde s_i\}_{i=2}^m$. We have that the share λ_τ for $\pi^\star(\tau)$ is

$$\lambda_\tau=\langle\mathbf{L}^\star_\tau,\mathbf{s}\rangle=\sum_{i\in[m]}L^\star_{\tau,i}sa^{i-1}+\sum_{i=2}^{m}L^\star_{\tau,i}\tilde s_i=\sum_{i\in[m]}L^\star_{\tau,i}sa^{i-1}+\tilde\lambda_\tau$$

for $\tau\in[\ell]$ where $\tilde\lambda_\tau=\sum_{i=2}^m L^\star_{\tau,i}\tilde s_i$ is known to $\mathcal{B}$. $\mathcal{B}$ then implicitly sets $t_\tau=-sb_\tau+\tilde t_\tau$ and computes $(C_{\tau,1},C_{\tau,2},C_{\tau,3})$ for each $\tau\in[\ell]$ as follows. There are two cases to distinguish.

- In the case of $\tau\in\mathsf{posi}$ (i.e., the τ-th row is associated with non-negated attribute), $\mathcal{B}$ computes $(C_{\tau,1},C_{\tau,2},C_{\tau,3})$ as follows:

$$
\begin{aligned}
C_{\tau,1}\ &=\ W^{\lambda_\tau}V^{t_\tau}=W^{\tilde\lambda_\tau}\cdot\prod_{i\in[m]}g^{L^\star_{\tau,i}sa^i}\cdot V^{\tilde{t_\tau}}\cdot\big(g^{\tilde v}\cdot\prod_{(j,k)\in\mathsf{posi}\times[m]}\big(g^{a^k/b_j}\big)^{L^\star_{j,k}}\big)^{-sb_\tau}\\
&=\ W^{\tilde\lambda_\tau}\cdot V^{\tilde t_\tau}\cdot(g^{sb_\tau})^{-\tilde v}\cdot\prod_{i\in[m]}g^{L^\star_{\tau,i}sa^i}\cdot\prod_{(j,k)\in\mathsf{posi}\times[m]}\big(g^{-sa^kb_\tau/b_j}\big)^{L^\star_{j,k}}
\end{aligned}
$$

$$
\begin{aligned}
&= W^{\tilde{\lambda}_\tau} \cdot V^{\tilde{t}_\tau} \cdot (g^{sb_\tau})^{-\tilde{v}} \cdot \prod_{i\in[m]} g^{L^\star_{\tau,i} sa^i} \cdot \prod_{k\in[m]} g^{-L^\star_{\tau,k} sa^k b_\tau/b_\tau} \cdot \prod_{(j,k)\in(\mathsf{posi}\setminus\{\tau\})\times[m]} \left(g^{sa^k b_\tau/b_j}\right)^{-L^\star_{j,k}} \\
&= W^{\tilde{\lambda}_\tau} \cdot V^{\tilde{t}_\tau} \cdot (g^{sb_\tau})^{-\tilde{v}} \cdot \prod_{(j,k)\in(\mathsf{posi}\setminus\{\tau\})\times[m]} \left(g^{sa^k b_\tau/b_j}\right)^{-L^\star_{j,k}}
\end{aligned}
$$

In the last equation above, the problematic terms $\{g^{sa^i}\}_{i=1}^m$ cancel out.

$$
\begin{aligned}
C_{\tau,2} &= (U^{\pi^\star(\tau)}H)^{-t_\tau} = (g^{sb_\tau})^{\tilde{u}\pi^\star(\tau)+\tilde{h}} \cdot (U^{\pi^\star(\tau)}H)^{-\tilde{t}_\tau} \cdot \prod_{(j,k)\in[\ell,m]} \left(g^{(\pi^\star(\tau)-\pi^\star(j))L^\star_{j,k}a^k/b_j^2}\right)^{sb_\tau} \\
&= (g^{sb_\tau})^{\tilde{u}\pi^\star(\tau)+\tilde{h}} \cdot (U^{\pi^\star(\tau)}H)^{-\tilde{t}_\tau} \cdot \prod_{\substack{(j,k)\in[\ell,m] \\ j\neq\tau}} \left(g^{sa^k b_\tau/b_j^2}\right)^{(\pi^\star(\tau)-\pi^\star(j))L^\star_{j,k}}
\end{aligned}
$$

In the equation above, the problematic terms $\{g^{sa^i/b_\tau}\}_{i=1}^m$ cancel out. Finally, $C_{\tau,3}$ can be computed as

$$C_{\tau,3} = (g^{sb_\tau})^{-1} \cdot g^{t_\tau}.$$

- In the case of $\tau \in \mathsf{nega}$ (i.e., the τ-th row is associated with negated attribute), $\mathcal{B}$ computes $(C_{\tau,1}, C_{\tau,2}, C_{\tau,3})$ as follows:

$$
\begin{aligned}
C_{\tau,1} &= W^{\lambda_\tau} \cdot (V')^{t_\tau} \\
&= W^{\tilde{\lambda}_\tau} \cdot \prod_{i\in[m]} g^{L^\star_{\tau,i} sa^i} \cdot (V')^{\tilde{t}_\tau} \\
&\quad \cdot \Big(\Big(\prod_{i\in\mathsf{nega}} g^{b_i} \Big)^{\tilde{u}} \cdot \prod_{\substack{(i,j,k)\in\mathsf{nega}\times[\ell,m] \\ i\neq j}} \left(g^{a^k b_i/b_j^2}\right)^{L^\star_{j,k}} \cdot \prod_{(j,k)\in\mathsf{nega}\times[m]} \left(g^{a^k/b_j}\right)^{L^\star_{j,k}} \Big)^{-sb_\tau} \\
&= \underbrace{W^{\tilde{\lambda}_\tau} \cdot (V')^{\tilde{t}_\tau} \cdot \Big(\prod_{i\in\mathsf{nega}} g^{sb_\tau b_i} \Big)^{-\tilde{u}}}_{\Phi_1} \cdot \prod_{\substack{(i,j,k)\in\mathsf{nega}\times[\ell,m] \\ i\neq j}} \left(g^{sa^k b_\tau b_i/b_j^2}\right)^{-L^\star_{j,k}} \\
&\quad \cdot \prod_{(j,k)\in\mathsf{nega}\times[m]} \left(g^{sa^k b_\tau/b_j}\right)^{-L^\star_{j,k}} \cdot \prod_{i\in[m]} g^{L^\star_{\tau,i} sa^i} \\
&= \Phi_1 \cdot \underbrace{\prod_{\substack{(i,j,k)\in\mathsf{nega}\times[\ell,m] \\ i\neq j, j\neq\tau}} \left(g^{sa^k b_\tau b_i/b_j^2}\right)^{-L^\star_{j,k}} \cdot \prod_{\substack{(i,k)\in\mathsf{nega}\times[m] \\ i\neq\tau}} \left(g^{sa^k b_i/b_\tau}\right)^{-L^\star_{\tau,k}}}_{\Phi_2} \\
&\quad \cdot \prod_{(j,k)\in\mathsf{nega}\times[m]} \left(g^{sa^k b_\tau/b_j}\right)^{-L^\star_{j,k}} \cdot \prod_{i\in[m]} g^{L^\star_{\tau,i} sa^i} \\
&= \Phi_1 \cdot \Phi_2 \cdot \prod_{(j,k)\in(\mathsf{nega}\setminus\{\tau\})\times[m]} \left(g^{sa^k b_\tau/b_j}\right)^{-L^\star_{j,k}} \cdot \prod_{k\in[m]} \left(g^{sa^k b_\tau/b_\tau}\right)^{-L^\star_{\tau,k}} \cdot \prod_{i\in[m]} g^{L^\star_{\tau,i} sa^i} \\
&= \Phi_1 \cdot \Phi_2 \cdot \prod_{(j,k)\in(\mathsf{nega}\setminus\{\tau\})\times[m]} \left(g^{sa^k b_\tau/b_j}\right)^{-L^\star_{j,k}}
\end{aligned}
$$

In the last equation above, the problematic terms $\{g^{sa^i}\}_{i=1}^m$ cancel out. Thus, $\mathcal{B}$ can compute $C_{\tau,1}$ efficiently using appropriate terms. $\mathcal{B}$ can also compute $C_{\tau,2}$ and $C_{\tau,3}$ by the same way as in the previous case.

Finally, $\mathcal{B}$ gives the challenge ciphertext $C = (C_0, C_1, \{C_{i,1}, C_{i,2}, C_{i,3}\}_{i\in[\ell]})$ to $\mathcal{A}$.

Guess. Finally, $\mathcal{A}$ outputs its guess β' for β. If $\beta' = \beta$, $\mathcal{A}$ outputs 1 for its guess. Otherwise, it outputs 0. If $T = e(g,g)^{sa^{n+1}}$, the above simulation is perfect and thus $\mathcal{A}$ has non-negligible advantage. On the other hand, If T is a random element in $\mathbb{G}_T$, $\mathcal{A}$'s advantage is 0. Therefore, if $\mathcal{A}$ breaks our scheme with non-negligible advantage, $\mathcal{B}$ has a non-negligible advantage in breaking the n-(B) assumption. □