A Generic Construction of CCA-secure Attribute-based Encryption with Equality Test *

Kyoichi Asano [†1], Keita Emura [†2], Atsushi Takayasu [†3], and Yohei Watanabe [†1]

[1] The University of Electro-Communications, Japan.
[2] Kanazawa University, Japan.
[3] The University of Tokyo, Japan.

April 2, 2024

Abstract

Attribute-based encryption with equality test (ABEET) is an extension of the ordinary attribute-based encryption (ABE), where trapdoors enable us to check whether two ciphertexts are encryptions of the same message. Thus far, several CCA-secure ABEET schemes have been proposed for monotone span programs satisfying selective security under q-type assumptions. In this paper, we propose a generic construction of CCA-secure ABEET from delegatable ABE. Specifically, our construction is an attribute-based extension of Lee et al.'s generic construction of identity-based encryption with equality test from hierarchical identity-based encryption. Even as far as we know, there are various delegatable ABE schemes. Therefore, we obtain various ABEET schemes with new properties that have not been achieved before such as various predicates, adaptive security, standard assumptions, compact ciphertexts/secret keys, and lattice-based constructions. To obtain several pairing-based ABEET schemes, we explicitly describe how to transform a pair encoding scheme to be delegatable. Moreover, we propose the first pair encoding scheme for key-policy ABE for non-monotone span programs with compact ciphertexts satisfying relaxed perfect security.

*An extended abstract appeared at ProvSec 2022 [AET+22]. This is the full version.

†During a part of this work, the authors are affiliated with National Institute of Information and Communications Technology, Japan.

Contents

1 Introduction

1.1 Background

The notion of public key encryption with equality test (PKEET) was introduced by Yang et al. [YTH+10]. PKEET is similar to public key encryption with keyword search [BCO+04, ABC+08] in a multi-user setting. PKEET has multiple public/secret key pairs $(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_N, \mathsf{sk}_N)$. Let ct_i and ct_j denote encryptions of plaintexts M_i and M_j by pk_i and pk_j, respectively. As the case of the standard public key encryption, the secret keys sk_i and sk_j can decrypt ct_i and ct_j, and recover M_i and M_j, respectively. Moreover, PKEET has a trapdoor td to perform the equality test. Let td_i and td_j denote trapdoors created by the secret keys sk_i and sk_j, respectively. Briefly speaking, even if the i-th user obtains the j-th trapdoor td_j, they cannot decrypt the j-th ciphertext ct_j. In contrast, any users who have trapdoors td_i and td_j can check whether ct_i and ct_j are encryptions of the same plaintexts. There are several applications of PKEET; for example, Yang et al. [YTH+10] considered outsourced databases with partitioning encrypted data where a database administrator can collect and categorize confidential data without help of message owners. Thus far, several PKEET schemes have been proposed [Tan11, LZL12, HTC+14, HTC+15, MZH+15, LLS+16a, LLS+16b, LSQ18, QYL+18, DFK+19, DSB+19, LLS+19, ZCZ+19, ZCL+19, LLS+20, LSQ+21] with stronger security models, efficiency improvements, additional properties, and under various assumptions.

As a natural extension of PKEET, attribute-based encryption with equality test (ABEET) has been studied. Here, we briefly explain ABEET with a predicate $\mathsf{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$. ABEET has a single master public/secret key pair $(\mathsf{mpk}, \mathsf{msk})$. Let ct_i and ct_j denote encryptions of plaintexts M_i and M_j for ciphertext-attributes x_i and x_j, respectively. As the case of the standard attribute-based encryption (ABE), the secret key sk_{y_i} for key attribute y_i (resp. sk_{y_j} for y_j) can decrypt ct_i (resp. ct_j) if $\mathsf{P}(x_i, y_i) = 1$ (resp. $\mathsf{P}(x_j, y_j) = 1$) holds. Let td_{y_i} and td_{y_j} denote trapdoors created by the secret keys sk_{y_i} and sk_{y_j}, respectively. Even if the user with the key-attribute y_i obtains the trapdoor td_{y_j} of the key-attribute y_j, they cannot decrypt the ciphertext ct_{x_j} of the ciphertext-attribute x_j when $\mathsf{P}(x_j, y_i) = 0$. In contrast, any users who have trapdoors td_{y_i} and td_{y_j} can check whether ct_{x_i} and ct_{x_j} are encryptions of the same plaintexts if $\mathsf{P}(x_i, y_i) = \mathsf{P}(x_j, y_j) = 1$ holds.

The simplest case of ABEET is arguably identity-based encryption with equality test (IBEET) that has an equality predicate $\mathsf{P}_{\mathsf{IBE}} : \mathcal{V} \times \mathcal{V} \to \{0,1\}$, i.e., $\mathsf{P}_{\mathsf{IBE}}(v, v') = 1 \Leftrightarrow v = v'$. Thus far, several IBEET schemes have been proposed such as [LLS+16b, Ma16, LSQ18, DLR+19, LMH+19, LLS+20, NSD+20, SDL20, LWS+21, AET24]. ABEET schemes for more complex monotone span programs have also been proposed [CHH+18, CHH+19, WCH+20, LSX+21] as ABE for the same predicate has been actively studied. However, ABEET research has a major drawback in the sense that progress in ABEET research is far behind that of ABE research. Although all the ABEET schemes [CHH+18, CHH+19, WCH+20, LSX+21] satisfy only selective security under q-type assumptions for monotone span programs, there are adaptively secure ABE schemes for monotone span programs under standard assumptions [LOS+10, Att14, Wee14, CGW15, Att16, CG17, Att19] and adaptively secure ABE schemes for more complex non-monotone span programs [AC17b, GWW19]. There are also several ABE schemes for other complex predicates such as (non-)deterministic finite automata [Att14, AC17b, GWW19, GW20] and circuits [BGG+14]. Although all the ABEET schemes [CHH+18, CHH+19, WCH+20, LSX+21] are pairing-based, there are lattice-based ABE schemes under the post-quantum learning with errors assumption such as [BGG+14]. Therefore, it is an important open problem to improve ABEET based on techniques of the state-of-the-art ABE schemes.

1.2 Our Contribution

To resolve the above mentioned open problem, we propose a generic construction of CCA-secure ABEET schemes from CPA-secure *delegatable* ABE schemes and cryptographic hash functions. To construct an ABEET scheme for a predicate $\mathsf{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$, our construction uses a delegatable ABE scheme with a hierarchical structure of the depth three, where only the first level supports the predicate $\mathsf{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ and the other two levels support only the equality predicate $\mathsf{P}_{\mathsf{IBE}} : \mathcal{V} \times \mathcal{V} \to \{0,1\}$. Since delegatable ABE has not been studied as much as (non-delegatable) ABE, our generic construction does not immediately provide ABEET schemes that have the same performance as all state-of-the-art ABE schemes. Nevertheless, there are several delegatable ABE schemes that enable us to obtain various more attractive ABEET schemes than known schemes [CHH+18, CHH+19, WCH+20, LSX+21]. At first, we can easily obtain selectively secure lattice-based ABEET schemes for circuits from Boneh et al.'s delegatable ABE scheme for circuits [BGG+14]. Next, we obtain several pairing-based ABEET schemes through the predicate encoding and pair encoding frameworks introduced by Wee [Wee14] and Attrapadung [Att14], respectively. These frameworks are unifying methods to design ABE for a large class of predicates, where the pair encoding can handle more complex predicates than the predicate encoding. Therefore, we can construct ABEET schemes for complex predicates captured by the predicate encoding and pair encoding frameworks. As a result, we obtain new and impressive ABEET schemes for various predicates at once.

Table 1 illustrates a comparison between CCA-secure ABEET schemes for some complex predicate including monotone span programs. All the schemes are constructed over prime-order bilinear groups. Since there are a huge number of ABE schemes through the pair encoding framework, all ABEET schemes obtained by our generic construction may not be covered in Table 1. However, 18 schemes listed in Table 1 should be sufficient for clarifying the impact of our generic construction. We briefly summarize how to obtain base ABE schemes as follows:

- Schemes 1 and 7: Instantiating predicate encoding scheme [Wee14] with compilers [CGW15, CG17].
- Schemes 2 and 8: Instantiating pair encoding scheme [Att14] with compilers [AC16a, Tak21].
- Scheme 3: Instantiating a pair encoding scheme in Section 5 with compilers [AC16a, Tak21].
- Scheme 9: Instantiating a pair encoding scheme [Tak21] with compilers [AC16a, Tak21].
- Schemes 4–6 and 10–12: Instantiating pair encoding schemes [Att19] with a compiler [AC17b].
- Schemes 13–18: Instantiating pair encoding schemes [Att14] with a compiler [AC17b].

Then, we explain various advantages of our results compared with known ABEET schemes for monotone span programs [CHH+18, CHH+19, WCH+20, LSX+21].

- Although all known ABEET schemes capture monotone span programs, Schemes 3–6 and 9–12 capture non-monotone span programs and Schemes 13–18 capture deterministic finite automata.
- Although all known ABEET schemes satisfy only selective security, Schemes 1, 2, 4–8, and 10–14 satisfy adaptive security and Schemes 3 and 9 satisfy semi-adaptive security.
- Although all known ABEET schemes except [LSX+21] support only small universe, Schemes 2–6 and 8–18 support large universe.

Table 1: Comparison among known CCA-secure ABEET schemes for complex predicates. MSP, NSP, DFA, CP, KP, ROM, and BDHE stand for monotone span program, non-monotone span program, deterministic finite automata, ciphertext-policy, key-policy, random oracle, and bilinear Diffie-Hellman exponent, respectively. The column "Compact Parameter" indicates that the content consists of the constant number of group elements.

Known Scheme	Predicate	Security	Policy	Universe	Model	Complexity Assumption	Compact Parameter
CHH+18 [CHH+18]	MSP	selective	CP	small	ROM	q-parallel BDHE	none
CHH+19 [CHH+19]	MSP	selective	CP	small	ROM	q-parallel BDHE	none
WCH+20 [WCH+20]	MSP	selective	CP	small	Std.	q-parallel BDHE	none
LSX+21 [LSX+21]	MSP	selective	CP	large	Std.	q-1	$\|\mathsf{mpk}\|$
Our Scheme (Base Schemes)	Predicate	Security	Policy	Universe	Model	Complexity Assumption	Compact Parameter
Scheme 1 ([Wee14, CGW15, CG17])	MSP	adaptive	KP	small	Std.	k-Lin	none
Scheme 2 ([Att14, AC16a, Tak21])	MSP	adaptive	KP	large	Std.	k-Lin	none
Scheme 3 ([AC16a, Tak21])	NSP	semi-adaptive	KP	large	Std.	k-Lin	$\|\mathsf{ct}\|$
Scheme 4 ([AC17b, Att19])	NSP	adaptive	KP	large	Std.	q-ratio	$\|\mathsf{mpk}\|$
Scheme 5 ([AC17b, Att19])	NSP	adaptive	KP	large	Std.	q-ratio	$\|\mathsf{ct}\|$
Scheme 6 ([AC17b, Att19])	NSP	adaptive	KP	large	Std.	q-ratio	$\|\mathsf{sk}\|$
Scheme 7 ([Wee14, CGW15, CG17])	MSP	adaptive	CP	small	Std.	k-Lin	none
Scheme 8 ([Att14, AC16a, Tak21])	MSP	adaptive	CP	large	Std.	k-Lin	none
Scheme 9 ([AC16a, Tak21])	NSP	semi-adaptive	CP	large	Std.	k-Lin	$\|\mathsf{ct}\|$
Scheme 10 ([AC17b, Att19])	NSP	adaptive	CP	large	Std.	q-ratio	$\|\mathsf{mpk}\|$
Scheme 11 ([AC17b, Att19])	NSP	adaptive	CP	large	Std.	q-ratio	$\|\mathsf{ct}\|$
Scheme 12 ([AC17b, Att19])	NSP	adaptive	CP	large	Std.	q-ratio	$\|\mathsf{sk}\|$
Scheme 13 ([Att14, AC17b])	DFA	adaptive	KP	large	Std.	q-ratio	$\|\mathsf{mpk}\|$
Scheme 14 ([Att14, AC17b])	DFA	adaptive	KP	large	Std.	q-ratio	$\|\mathsf{ct}\|$
Scheme 15 ([Att14, AC17b])	DFA	adaptive	KP	large	Std.	q-ratio	$\|\mathsf{sk}\|$
Scheme 16 ([Att14, AC17b])	DFA	adaptive	CP	large	Std.	q-ratio	$\|\mathsf{mpk}\|$
Scheme 17 ([Att14, AC17b])	DFA	adaptive	CP	large	Std.	q-ratio	$\|\mathsf{ct}\|$
Scheme 18 ([Att14, AC17b])	DFA	adaptive	CP	large	Std.	q-ratio	$\|\mathsf{sk}\|$

- Although security of all known ABEET schemes are based on q-type assumptions, security of Schemes 1–3 and 7–9 are based on the standard k-linear assumption.
- Although all known ABEET schemes do not have compact ciphertexts and secret keys, Schemes 3, 5, 9, 11, 14, and 17 have compact ciphertexts and Schemes 6, 12, 15, and 18 have compact secret keys.

Therefore, we successfully obtain several improved ABEET schemes from our generic construction. Moreover, although we only list proposed ABEET schemes for complex predicates in Table 1, our generic construction also provides various ABEET schemes for less expressive but important predicates captured by the pair encoding and the predicate encoding such as (non-zero) inner product encryption, (negated) spatial encryption, doubly spatial encryption, and arithmetic span programs.

1.3 Technical Overview

We explain an overview of our construction. At first, we exploit the common essence of known ABEET constructions and briefly summarize the fact that any IND-CPA secure ABE scheme for a predicate $\mathsf{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ becomes CPA-secure ABEET scheme for the same predicate by combining with cryptographic hash functions. For this purpose, we run two ABE schemes for the same predicate in parallel. Let $\mathsf{ABE.mpk}_0$ and $\mathsf{ABE.mpk}_1$ denote master public keys of the two ABE schemes and let H denote a cryptographic hash function. Then, we set $\mathsf{mpk} = (\mathsf{ABE.mpk}_0, \mathsf{ABE.mpk}_1, \mathsf{H})$ as the master public key of an ABEET scheme. We encrypt a plaintext M for a ciphertext attribute $x \in \mathcal{X}$ as $\mathsf{ct}_x = (\mathsf{ABE.ct}_{x,0}, \mathsf{ABE.ct}_{x,1})$, where $\mathsf{ABE.ct}_{x,0}$ and $\mathsf{ABE.ct}_{x,1}$ are encryptions of M and $\mathsf{H}(\mathsf{M})$ for the same x computed by $\mathsf{ABE.mpk}_0$ and $\mathsf{ABE.mpk}_1$, respectively. We set a secret key of a key attribute $y \in \mathcal{Y}$ as $\mathsf{sk}_y = (\mathsf{ABE.sk}_{y,0}, \mathsf{ABE.sk}_{y,1})$, where $\mathsf{ABE.sk}_{y,0}$ and $\mathsf{ABE.sk}_{y,1}$ are secret keys for the same y computed by $(\mathsf{ABE.mpk}_0, \mathsf{ABE.msk}_0)$ and $(\mathsf{ABE.mpk}_1, \mathsf{ABE.msk}_1)$, respectively. The secret key sk_y can decrypt the ciphertext ct_x if $\mathsf{P}(x,y) = 1$ by simply decrypting the ABE ciphertext $\mathsf{ABE.ct}_{x,0}$ with the ABE secret key $\mathsf{ABE.sk}_{y,0}$ and recover M. We set a trapdoor for $y \in \mathcal{Y}$ as $\mathsf{td}_y = \mathsf{ABE.sk}_{y,1}$. Given two ciphertexts $(\mathsf{ct}_x, \mathsf{ct}_{x'})$ for $(x, x') \in \mathcal{X}^2$ and two trapdoors $(\mathsf{td}_y, \mathsf{td}_{y'})$ such that $\mathsf{P}(x,y) = \mathsf{P}(x',y') = 1$, we can check whether the two ciphertexts are encryptions of the same plaintexts by checking whether the decryption results of the ABE ciphertexts $\mathsf{ABE.ct}_{x,1}$ and $\mathsf{ABE.ct}_{x',1}$ by the trapdoors $\mathsf{ABE.sk}_{y,1}$ and $\mathsf{ABE.sk}_{y',1}$, respectively, have the same values.

Next, we observe that the above ABEET scheme satisfies CPA security. Briefly speaking, ABEET has to be secure against two types of adversaries called Type-I and Type-II. Let x^* denote the target ciphertext attribute. The Type-I adversary can receive trapdoors td_y such that $\mathsf{P}(x^*, y) = 1$, while the Type-II adversary cannot receive such trapdoors. Although the Type-I adversary trivially breaks indistinguishability by definition, we can prove one-wayness against the Type-I adversary. Thus, the challenge ciphertext $\mathsf{ct}^*_{x^*}$ is an encryption of M^* that is sampled uniformly at random from the plaintext space. The IND-CPA security of the underlying ABE scheme ensures that the first element $\mathsf{ABE.ct}^*_{x^*,0}$ of the challenge ciphertext $\mathsf{ct}^*_{x^*}$ does not reveal the information of M^* at all. Since the Type-I adversary has the trapdoor $\mathsf{td}_y = \mathsf{ABE.sk}_{y,1}$ such that $\mathsf{P}(x^*, y) = 1$, it can recover $\mathsf{H}(\mathsf{M}^*)$; however, the one-wayness of the hash function H ensures that M^* cannot be recovered. In contrast, we have to prove indistinguishability against the Type-II adversary. Thus, the challenge ciphertext $\mathsf{ct}^*_{x^*}$ is an encryption of $\mathsf{M}^*_{\mathsf{coin}}$, where the tuple $(\mathsf{M}^*_0, \mathsf{M}^*_1)$ is declared by the adversary and $\mathsf{coin} \leftarrow_{\$} \{0,1\}$ is flipped by the challenger. In this case, the IND-CPA security of the underlying ABE scheme ensures that both $\mathsf{ABE.ct}^*_{x^*,0}$ and $\mathsf{ABE.ct}^*_{x^*,1}$ do not reveal the information of $\mathsf{M}^*_{\mathsf{coin}}$ and $\mathsf{H}(\mathsf{M}^*_{\mathsf{coin}})$ at all, respectively. We note that the above construction does not provide CCA security even if the underlying ABE scheme satisfies IND-CCA security. Indeed, when the Type-II adversary receives the challenge ciphertext $\mathsf{ct}^*_{x^*} = (\mathsf{ABE.ct}^*_{x^*,0}, \mathsf{ABE.ct}^*_{x^*,1})$, it can guess the value of coin by making a decryption query on $(\mathsf{ABE.ct}_{x^*,0}, \mathsf{ABE.ct}^*_{x^*,1})$, where $\mathsf{ABE.ct}_{x^*,0}$ is the encryption of M^*_0 or M^*_1 computed by the adversary itself.

Based on the discussion so far, what we have to achieve is CCA security. For this purpose, we follow the generic construction of CCA-secure IBEET from IND-CPA secure hierarchical IBE with the depth three proposed by Lee et al. [LLS+20]. Lee et al. used the CHK transformation [CHK04] to update the above scheme for achieving CCA security in the identity-based setting. Similarly, we use the Yamada et al.'s transformation [YAH+11], which is the attribute-based variant of the CHK transformation, to update the above CPA-secure construction for achieving CCA security in the attribute-based setting. We use a IND-CPA-secure delegatable ABE scheme with the depth three as a building block. Specifically, to construct ABEET for a predicate $\mathsf{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$, we use a delegatable ABE scheme for a predicate $(\mathcal{X} \times \{0,1\} \times \mathcal{V}) \times (\mathcal{Y} \times \{0,1\} \times \mathcal{V}) \to \{0,1\}$, where a secret key $\mathsf{ABE.sk}_{y,b',v'}$ can decrypt a ciphertext $\mathsf{ABE.ct}_{x,b,v}$ correctly if it holds that

$\mathsf{P}(x, y) = 1 \wedge b = b' \wedge v = v'$. Here, we use the second hierarchical level $b, b' \in \{0, 1\}$ to specify which of the ABE schemes in the above CPA-secure construction and the third level $v, v' \in \mathcal{V}$ to specify verification keys of the one-time signature scheme. As a result, we set a master public key, ciphertexts for $x \in \mathcal{X}$, secret keys and trapdoors for $y \in \mathcal{Y}$ of ABEET as $\mathsf{mpk} = \mathsf{ABE.mpk}$, $\mathsf{ct}_x = (\mathsf{verk}, \mathsf{ABE.ct}_{x,0,\mathsf{verk}}, \mathsf{ABE.ct}_{x,1,\mathsf{verk}}, \sigma)$, $\mathsf{sk}_y = \mathsf{ABE.sk}_y$, and $\mathsf{td}_y = \mathsf{ABE.sk}_{y,1}$, respectively, where verk is a verification key of the one-time signature scheme and σ is a signature for the message $[\mathsf{ABE.ct}_{x,0,\mathsf{verk}} \| \mathsf{ABE.ct}_{x,1,\mathsf{verk}}]$. Intuitively, the construction achieves CCA security by combining with security of the above CPA-secure construction and Yamada et al.'s technique [YAH+11].

1.4 Difference from the Conference Version [AET+22]

From the preliminary version of this paper [AET+22], this full version contains three main updates summarized as follows.

Delegatable Transformation for Pair Encoding Scheme. As we explained in Section 1.3, we do not use ABE itself but its delegatable one to construct ABEET. In other words, if we want to construct ABEET for expressive predicates P, we have to construct delegatable ABE schemes whose first hierarchical level supports P. In the case of lattice-based constructions, Boneh et al. [BGG+14] constructed delegatable ABE schemes for circuits. If we do not consider expressive predicates for pairing-based constructions, Ambrona et al. [ABS17] proposed a transformation for predicate encoding schemes to be delegatable ones. In contrast, there are no corresponding transformations for pair encoding schemes that can handle more complex predicates than predicate encoding schemes. In the preliminary version of this paper, we claimed that the desired transformation was available by extending the Ambrona et al.'s transformation; however, we did not describe a concrete transformation. In this full version, we explicitly describe how to transform pair encoding schemes to be delegatable ones in Section 5.2. Although a restricted property of delegatable ABE is sufficient for constructing ABEET as we explained in Section 1.3, our proposed delegatable transformation is general in the sense that we can handle an arbitrary number of arbitrary predicates as long as there are pair encoding schemes for these predicates.

New Pair Encoding Scheme. In the preliminary version of this paper, Scheme 3 in Table 1 was a key-policy ABEET scheme for *monotone* span programs with compact ciphertexts. We obtained the scheme from Agrawal and Chase's relaxed perfectly secure pair encoding scheme[1] [AC16a] for the same predicate with compilers [AC16a, Tak21]. In contrast, we propose a new relaxed perfectly secure pair encoding scheme for key-policy ABE for *non-monotone* span programs with compact ciphertexts in Section 5.3. From the pair encoding scheme with compilers [AC16a, Tak21], Scheme 3 in Table 1 supports non-monotone span programs. Moreover, although our pair encoding scheme supports more complex non-monotone predicates, the proposed pair encoding scheme is more efficient than Agrawal and Chase's one.

New Instantiations of ABEET for DFA. In the preliminary version of this paper, there are only two ABEET schemes for DFA, i.e., Schemes 13 and 16 in Table 1. We obtained the scheme from Attrapadung's pair encoding schemes [Att14] with a compiler [AC17b]. Since the pair encoding schemes satisfy symbolic security introduced in [AC17b], we applied Agrawal and Chase's transformation for symbolically secure pair encoding schemes to be those with compact ciphertexts/secret keys. As a result, we obtain Schemes 14, 15, 17, and 18 in this full version.

[1] To be precise, the pair encoding scheme itself was introduced by Attrapadung [Att14]; however, its instantiation requires a complex q-type assumption. Afterwards, Agrawal and Chase proved that the pair encoding scheme satisfies relaxed perfect security; therefore, its instantiation requires only the standard k-linear assumption.

1.5 Roadmap

In Section 2, we introduce notations and give some definitions. We show our generic construction of ABEET and prove its correctness in Section 3. We provide security proofs of our construction in Section 4. In Section 5, we propose a transformation for a pair encoding scheme to be delegatable and a new pair encoding scheme for key-policy ABE for non-monotone span programs.

2 Preliminaries

Notation. Throughout the paper, λ denotes a security parameter. For an i-bit binary string $\mathsf{s}_1 \in \{0,1\}^i$ and a j-bit binary string $\mathsf{s}_2 \in \{0,1\}^j$, let $[\mathsf{s}_1 \| \mathsf{s}_2] \in \{0,1\}^{i+j}$ denote an $(i+j)$-bit concatenation of s_1 and s_2. For a finite set S, $s \leftarrow_{\$} S$ denotes a sampling of an element s from S uniformly at random and let $|S|$ denotes a cardinality of S. Probabilistic polynomial time is abbreviated as PPT. For two probability distributions, "$\equiv$" and "$\approx$" denote the same distribution and statistically indistinguishable, respectively. Let bold letters $\mathbf{a}$ and $\mathbf{A}$ denote a row vector and a matrix, respectively.

2.1 Delegatable Attribute-based Encryption

We define delegatable ABE (or simply called ABE hereafter). To make readers easier to understand, we here consider a special case of ABE, which is sufficient to describe our construction. The definition we use here differs from the general definition of ABE in the following ways:

- The hierarchical level is three, not an arbitrary number.
- The second and third levels support only the equality predicate as in identity-based encryption, where the second level and third level take elements of $\{0,1\}$ and an identity space $\mathcal{V}$, respectively.
- The Enc algorithm always takes a level-3 attribute.

Let $\mathsf{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ denotes a predicate, where $\mathcal{X}$ and $\mathcal{Y}$ are attribute spaces for ciphertexts and secret keys, respectively. In our definition of ABE for a predicate P, ciphertexts $\mathsf{ABE.ct}_{x,b,v}$ and secret keys $\mathsf{ABE.sk}_{y,b',v'}$ are associated with $(x,b,v) \in \mathcal{X} \times \{0,1\} \times \mathcal{V}$ and $(y,b',v') \in \mathcal{Y} \times \{0,1\} \times \mathcal{V}$, respectively. A secret key $\mathsf{ABE.sk}_{y,b',v'}$ can decrypt a ciphertext $\mathsf{ABE.ct}_{x,b,v}$ if it holds that $\mathsf{P}(x,y) = 1 \wedge b = b' \wedge v = v'$.

Syntax. An ABE scheme Π_{ABE} for a predicate P consists of the five algorithms (ABE.Setup, ABE.KeyGen, ABE.Enc, ABE.Dec, ABE.Delegate) as follows:

$\mathsf{ABE.Setup}(1^\lambda) \to (\mathsf{ABE.mpk}, \mathsf{ABE.msk})$**:** On input the security parameter 1^λ, it outputs a master public key ABE.mpk and a master secret key ABE.msk. We assume that ABE.mpk contains a description of a plaintext space $\mathcal{M}$ that is determined only by the security parameter λ.

$\mathsf{ABE.Enc}(\mathsf{ABE.mpk}, (x,b,v), \mathsf{M}) \to \mathsf{ABE.ct}_{x,b,v}$**:** On input a master public key ABE.mpk, $(x,b,v) \in \mathcal{X} \times \{0,1\} \times \mathcal{V}$, and a plaintext $\mathsf{M} \in \mathcal{M}$, it outputs a ciphertext $\mathsf{ABE.ct}_{x,b,v}$.

$\mathsf{ABE.KeyGen}(\mathsf{ABE.mpk}, \mathsf{ABE.msk}, Y) \to \mathsf{ABE.sk}_Y$**:** On input a master public key ABE.mpk, a master secret key ABE.msk, and Y, it outputs a secret key $\mathsf{ABE.sk}_Y$, where Y is the element of $\mathcal{Y}$, $\mathcal{Y} \times \{0,1\}$ or $\mathcal{Y} \times \{0,1\} \times \mathcal{V}$.

$\mathsf{ABE.Dec}(\mathsf{ABE.mpk}, \mathsf{ABE.ct}_{x,b,v}, \mathsf{ABE.sk}_{y,b',v'}) \to \mathsf{M}$ or $\perp$: On input a master public key $\mathsf{ABE.mpk}$, a ciphertext $\mathsf{ABE.ct}_{x,b,v}$, and a secret key $\mathsf{ABE.sk}_{y,b',v'}$, it outputs the decryption result M if $\mathsf{P}(x,y) = 1 \wedge (b,v) = (b',v')$. Otherwise, output $\perp$.

$\mathsf{ABE.Delegate}(\mathsf{ABE.mpk}, \mathsf{ABE.sk}_Y, Y') \to \mathsf{ABE.sk}_{Y'}$**:** On input a master public key $\mathsf{ABE.mpk}$, a secret key $\mathsf{ABE.sk}_Y$ and Y', it outputs a secret key $\mathsf{ABE.sk}_{Y'}$, where Y is the element of $\mathcal{Y}$ or $\mathcal{Y} \times \{0,1\}$, Y' is the element of $\{Y\} \times \{0,1\}$ or $\{Y\} \times \{0,1\} \times \mathcal{V}$ if $Y \in \mathcal{Y}$, and Y' is the element of $\{Y\} \times \{0,1\} \times \mathcal{V}$ if $Y \in \mathcal{Y} \times \{0,1\}$.

Correctness. For all $\lambda \in \mathbb{N}$, all $(\mathsf{ABE.mpk}, \mathsf{ABE.msk}) \leftarrow \mathsf{ABE.Setup}(1^\lambda)$, all $\mathsf{M} \in \mathcal{M}$, all $(x,y) \in \mathcal{X} \times \mathcal{Y}$ such that $\mathsf{P}(x,y) = 1$, and all $(b,v) \in \{0,1\} \times \mathcal{V}$, it is required that $\mathsf{M}' = \mathsf{M}$ holds with overwhelming probability, where $\mathsf{ABE.ct}_{x,b,v} \leftarrow \mathsf{ABE.Enc}(\mathsf{ABE.mpk}, (x,b,v), \mathsf{M})$, $\mathsf{ABE.sk}_{y,b,v} \leftarrow \mathsf{ABE.KeyGen}(\mathsf{ABE.mpk}, \mathsf{ABE.msk}, (y,b,v))$, and $\mathsf{M}' \leftarrow \mathsf{ABE.Dec}(\mathsf{ABE.mpk}, \mathsf{ABE.ct}_{x,b,v}, \mathsf{ABE.sk}_{y,b,v})$. In addition, there is a correctness for $\mathsf{ABE.Delegate}$, where outputs of $\mathsf{ABE.KeyGen}(\mathsf{ABE.mpk}, \mathsf{ABE.msk}, Y')$ and $\mathsf{ABE.Delegate}(\mathsf{ABE.mpk}, \mathsf{ABE.KeyGen}(\mathsf{ABE.mpk}, \mathsf{ABE.msk}, Y), Y')$ follow the same distribution.

Security. We consider adaptive IND-CPA security defined below. Note that the following definition is specific to the above syntax but implied by the general adaptive IND-CPA definition.

Definition 2.1 (Adaptive IND-CPA Security)**.** The adaptive IND-CPA security of an ABE scheme Π_{ABE} is defined by a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ as follows:

Init: $\mathcal{C}$ runs $(\mathsf{ABE.mpk}, \mathsf{ABE.msk}) \leftarrow \mathsf{ABE.Setup}(1^\lambda)$ and gives $\mathsf{ABE.mpk}$ to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ is allowed to make the following key extraction queries to $\mathcal{C}$:

Key extraction query: $\mathcal{A}$ is allowed to make the query on Y. Upon the query, $\mathcal{C}$ runs $\mathsf{ABE.sk}_Y \leftarrow \mathsf{ABE.KeyGen}(\mathsf{ABE.mpk}, \mathsf{ABE.msk}, Y)$ and returns $\mathsf{ABE.sk}_Y$ to $\mathcal{A}$, where Y is the element of $\mathcal{Y}$, $\mathcal{Y} \times \{0,1\}$ or $\mathcal{Y} \times \{0,1\} \times \mathcal{V}$.

Challenge query: $\mathcal{A}$ is allowed to make the query only once. Upon $\mathcal{A}$'s query on $((x^*, b^*, v^*), {\mathsf{M}_0}^*, {\mathsf{M}_1}^*) \in \mathcal{X} \times \{0,1\} \times \mathcal{V} \times \mathcal{M}^2$, where ${\mathsf{M}_0}^*$ and ${\mathsf{M}_1}^*$ have the same length and (x^*, b^*, v^*) should not satisfy the following conditions for all the attributes Y queried on key extraction queries in Phase 1:

- If $Y = y \in \mathcal{Y}$, $\mathsf{P}(x^*, y) = 1$ holds.
- If $Y = (y,b) \in \mathcal{Y} \times \{0,1\}$, $\mathsf{P}(x^*, y) = 1 \wedge b^* = b$ holds.
- If $Y = (y,b,v) \in \mathcal{Y} \times \{0,1\} \times \mathcal{V}$, $\mathsf{P}(x^*, y) = 1 \wedge (b^*, v^*) = (b, v)$ holds.

Then, $\mathcal{C}$ flips a coin $\mathsf{coin} \leftarrow_{\$} \{0,1\}$ and runs $\mathsf{ABE.ct}^*_{x^*,b^*,v^*} \leftarrow \mathsf{ABE.Enc}(\mathsf{ABE.mpk}, (x^*, b^*, v^*), \mathsf{M}^*_{\mathsf{coin}})$. Then, $\mathcal{C}$ returns $\mathsf{ABE.ct}^*_{x^*,b^*,v^*}$ to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ is allowed to make key extraction queries as in Phase 1 with the following exceptions:

Key extraction query: Upon $\mathcal{A}$'s query on Y, Y should not satisfy the conditions with x^* as we mentioned in the challenge query.

Guess: At the end of the game, $\mathcal{A}$ returns $\widehat{\mathsf{coin}} \in \{0,1\}$ as a guess of coin.

The adversary $\mathcal{A}$ wins in the above game if $\widehat{\mathsf{coin}} = \mathsf{coin}$ and the advantage is defined to

$$\mathsf{Adv}^{\text{IND-CPA}}_{\Pi_{\mathsf{ABE}},\mathcal{A}}(\lambda) := \left|\Pr\left[\widehat{\mathsf{coin}} = \mathsf{coin}\right] - \frac{1}{2}\right|.$$

If $\mathsf{Adv}^{\text{IND-CPA}}_{\Pi_{\mathsf{ABE}},\mathcal{A}}(\lambda)$ is negligible in the security parameter λ for all PPT adversaries $\mathcal{A}$, an ABE scheme Π_{ABE} is said to satisfy adaptive IND-CPA security.

Remark 1. The Definition 2.1 states the adaptive IND-CPA security in the sense that $\mathcal{A}$ declares the target (x^*, b^*, v^*) at the challenge query. The *selective* IND-CPA security can be defined in the same way except that $\mathcal{A}$ declares the target (x^*, b^*, v^*) before the init phase. Similarly, the *semi-adaptive* IND-CPA security can be defined in the same way except that $\mathcal{A}$ declares the target (x^*, b^*, v^*) just after the init phase.

2.2 One-time Signature

Syntax. An one-time signature (OTS) scheme Γ consists of three algorithms ($\mathsf{Sig.Setup}$, $\mathsf{Sig.Sign}$, $\mathsf{Sig.Vrfy}$) with the same message space $\mathcal{M}$ used in IBE scheme as follows:

$\mathsf{Sig.Setup}(1^\lambda) \to (\mathsf{verk}, \mathsf{sigk})$**:** On input the security parameter 1^λ, it outputs a verification key verk and signing key sigk.

$\mathsf{Sig.Sign}(\mathsf{sigk}, \mathsf{M}) \to \sigma$**:** On input a signing key sigk and a message $\mathsf{M} \in \mathcal{M}$, it outputs a signature σ.

$\mathsf{Sig.Vrfy}(\mathsf{verk}, \mathsf{M}, \sigma) \to 1$ or 0**:** On input a verification key verk, a message $\mathsf{M} \in \mathcal{M}$, and its signature σ, it outputs 1 if the signature is valid and outputs 0 $\perp$ otherwise.

Correctness. We require that for all security parameters $\lambda \in \mathbb{N}$, $(\mathsf{verk}, \mathsf{sigk}) \leftarrow \mathsf{Sig.Setup}(1^\lambda)$, and messages $\mathsf{M} \in \{0,1\}^*$, it holds that $\mathsf{Sig.Vrfy}(\mathsf{verk}, \mathsf{M}, \mathsf{Sig.Sign}(\mathsf{sigk}, \mathsf{M})) = 1$ with overwhelming probability.

Security. We define a security notion for OTS. Let Γ be an OTS scheme, and we consider a game between an adversary $\mathcal{A}$ and the challenger $\mathcal{C}$. The game is parameterized by the security parameter λ. The game proceeds as follows: $\mathcal{C}$ first runs $(\mathsf{verk}, \mathsf{sigk}) \leftarrow \mathsf{Sig.Setup}(1^\lambda)$ and gives verk to $\mathcal{A}$. $\mathcal{A}$ is allowed to make the *signature generation query* only once: upon a query $\mathsf{M} \in \{0,1\}^*$ from $\mathcal{A}$, $\mathcal{C}$ returns $\sigma \leftarrow \mathsf{Sig.Sign}(\mathsf{sigk}, \mathsf{M})$ to $\mathcal{A}$. $\mathcal{A}$ outputs $(\widehat{\mathsf{M}}, \hat{\sigma})$ and terminates. In this game, $\mathcal{A}$'s advantage is defined by

$$\mathsf{Adv}^{\mathsf{OTS}}_{\Gamma,\mathcal{A}}(\lambda) := \Pr[\mathsf{Sig.Vrfy}(\mathsf{verk}, \widehat{\mathsf{M}}, \hat{\sigma}) \to 1 \wedge (\widehat{\mathsf{M}}, \hat{\sigma}) \neq (\mathsf{M}, \sigma)].$$

Definition 2.2 (Strong Unforgeability)**.** We say that an OTS scheme Γ satisfies strong unforgeability, if the advantage $\mathsf{Adv}^{\mathsf{OTS}}_{\Gamma,\mathcal{A}}(\lambda)$ is negligible for all PPT adversaries $\mathcal{A}$.

2.3 Hash Functions

Let $\mathsf{H} : \mathcal{M} \to \mathcal{R}$ be a hash function. We require the following properties of hash functions for our schemes.

Definition 2.3 (One-wayness)**.** We say that a hash function H is one-way (or preimage resistant) if for all PPT adversaries $\mathcal{A}$,

$$\mathsf{Adv}^{\mathrm{OW}}_{\mathsf{H},\mathcal{A}}(\lambda) := \Pr\left[\mathsf{M}^* \leftarrow_{\$} \mathcal{M}, \widehat{\mathsf{M}} \leftarrow \mathcal{A}(\mathsf{H}(\mathsf{M}^*)) : \mathsf{H}(\widehat{\mathsf{M}}) = \mathsf{H}(\mathsf{M}^*)\right]$$

is negligible in λ.

Definition 2.4 (Collision Resistance)**.** We say that a hash function H is collision resistant if for all PPT adversaries $\mathcal{A}$,

$$\mathsf{Adv}^{\mathrm{CR}}_{\mathsf{H},\mathcal{A}}(\lambda) := \Pr[(\mathsf{M}_0, \mathsf{M}_1) \leftarrow \mathcal{A} : \mathsf{M}_0 \neq \mathsf{M}_1 \wedge \mathsf{H}(\mathsf{M}_0) = \mathsf{H}(\mathsf{M}_1)]$$

is negligible in λ.

2.4 Attribute-based Encryption with Equality Test

Syntax. An ABEET scheme Π for a predicate $\mathsf{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ consists of the following six algorithms (Setup, Enc, KeyGen, Dec, $\mathsf{Trapdoor}$, Test) as follows:

$\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$**:** On input the security parameter 1^λ, it outputs a master public key mpk and a master secret key msk. We assume that mpk contains a description of a plaintext space $\mathcal{M}$ that is determined only by the security parameter λ.

$\mathsf{Enc}(\mathsf{mpk}, x, \mathsf{M}) \to \mathsf{ct}_x$**:** On input a master public key mpk, $x \in \mathcal{X}$, and a plaintext $\mathsf{M} \in \mathcal{M}$, it outputs a ciphertext ct_x.

$\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, y) \to \mathsf{sk}_y$**:** On input a master public key mpk, a master secret key msk, and $y \in \mathcal{Y}$, it outputs a secret key sk_y.

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{ct}_x, \mathsf{sk}_y) \to \mathsf{M}$ or $\perp$**:** On input a master public key mpk, a ciphertext ct_x, and a secret key sk_y, it outputs the decryption result M if $\mathsf{P}(x, y) = 1$. Otherwise, output $\perp$.

$\mathsf{Trapdoor}(\mathsf{mpk}, \mathsf{sk}_y) \to \mathsf{td}_y$**:** On input a master public key mpk and a secret key sk_y, it outputs the trapdoor td_y for $y \in \mathcal{Y}$.

$\mathsf{Test}(\mathsf{ct}_x, \mathsf{td}_y, \mathsf{ct}_{x'}, \mathsf{td}_{y'}) \to 1$ or 0**:** On input two ciphertexts $\mathsf{ct}_x, \mathsf{ct}_{x'}$ and two trapdoors td_y, $\mathsf{td}_{y'}$, it outputs 1 or 0.

Correctness. We require an ABEET scheme to satisfy the following three conditions. Briefly speaking, the first condition ensures that the Dec algorithm works correctly. In contrast, the second (resp. third) conditions ensure that the Test algorithm outputs 1 (resp. 0) if ct_x and $\mathsf{ct}_{x'}$ are encryptions of the same plaintext (resp. distinct plaintexts), respectively. We consider PPT adversaries for the third condition. The three conditions are formally defined as follows:

(1) For all $\lambda \in \mathbb{N}$, all $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$, all $\mathsf{M} \in \mathcal{M}$, all $x \in \mathcal{X}$ and all $y \in \mathcal{Y}$, such that $\mathsf{P}(x, y) = 1$, it is required that $\mathsf{M}' = \mathsf{M}$ holds with overwhelming probability, where $\mathsf{ct}_x \leftarrow \mathsf{Enc}(\mathsf{mpk}, x, \mathsf{M})$, $\mathsf{sk}_y \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, y)$, and $\mathsf{M}' \leftarrow \mathsf{Dec}(\mathsf{mpk}, \mathsf{ct}_x, \mathsf{sk}_y)$.

(2) For all $\lambda \in \mathbb{N}$, all $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$, all $\mathsf{M} \in \mathcal{M}$, all $x_0, x_1 \in \mathcal{X}$ and all $y_0, y_1 \in \mathcal{Y}$, such that $\wedge_{i \in \{0,1\}} \mathsf{P}(x_i, y_i) = 1$, it is required that $1 \leftarrow \mathsf{Test}(\mathsf{ct}_{x_0}, \mathsf{td}_{y_0}, \mathsf{ct}_{x_1}, \mathsf{td}_{y_1})$ holds with overwhelming probability, where $\mathsf{sk}_{y_i} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, y_i)$, $\mathsf{ct}_{x_i} \leftarrow \mathsf{Enc}(\mathsf{mpk}, x_i, \mathsf{M})$, and $\mathsf{td}_{y_i} \leftarrow \mathsf{Trapdoor}(\mathsf{mpk}, \mathsf{sk}_{y_i})$ for $i = 0, 1$.

(3) For all $\lambda \in \mathbb{N}$, all $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$, all PPT adversaries $\mathcal{A}$, all $x_0, x_1 \in \mathcal{X}$ and all $y_0, y_1 \in \mathcal{Y}$, such that $\wedge_{i\in\{0,1\}}\mathsf{P}(x_i, y_i) = 1$, it is required that

$$\mathsf{M}_0 \neq \mathsf{M}_1 \wedge 1 \leftarrow \mathsf{Test}(\mathsf{mpk}, \mathsf{ct}_{x_0}, \mathsf{td}_{y_0}, \mathsf{ct}_{x_1}, \mathsf{td}_{y_1})$$

holds with negligible probability, where $(\mathsf{M}_0, \mathsf{M}_1) \leftarrow \mathcal{A}(\mathsf{mpk}, \mathsf{msk})$, $\mathsf{sk}_{y_i} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, y_i)$, $\mathsf{ct}_{x_i} \leftarrow \mathsf{Enc}(\mathsf{mpk}, x_i, \mathsf{M}_i)$, and $\mathsf{td}_{y_i} \leftarrow \mathsf{Trapdoor}(\mathsf{mpk}, \mathsf{sk}_{y_i})$ for $i = 0, 1$.

Remark 2. In most ABEET papers, PPT adversaries do not appear in the definition of the third condition. In these works, the authors defined the third condition in the same way as the second condition except that $0 \leftarrow \mathsf{Test}(\mathsf{ct}_{x_0}, \mathsf{td}_{y_0}, \mathsf{ct}_{x_1}, \mathsf{td}_{y_1})$ holds with overwhelming probability, where $\mathsf{ct}_{x_0} \leftarrow \mathsf{Enc}(\mathsf{mpk}, x_0, \mathsf{M}_0)$ and $\mathsf{ct}_{x_1} \leftarrow \mathsf{Enc}(\mathsf{mpk}, x_1, \mathsf{M}_1)$ such that $\mathsf{M}_0 \neq \mathsf{M}_1$. Then, the authors proved the third condition based on the collision resistance of hash functions. However, the collision resistance itself is insufficient for proving the condition because unbounded adversaries may be able to find collisions. To this end, we modify the definition along with PPT adversaries and formally prove the condition based on the collision resistance of hash functions.

Security. For the security of ABEET, we consider two different types of adversaries. One has a trapdoor for the target attribute or not.

- Type-I adversary: This type of adversaries has trapdoors td_y such that $\mathsf{P}(x^*, y) = 1$. Therefore, the adversaries can perform the equality test with the challenge ciphertext $\mathsf{ct}^*_{x^*}$. Hence, we consider one-wayness.
- Type-II adversary: This type of adversaries has no trapdoors td_y such that $\mathsf{P}(x^*, y) = 1$. Therefore, the adversaries cannot perform the equality test with the challenge ciphertext $\mathsf{ct}^*_{x^*}$. Hence, we consider indistinguishability.

Definition 2.5 (Adaptive OW-CCA2 Security against Type-I Adversaries)**.** The adaptive OW-CCA2 security against Type-I adversaries of an ABEET scheme Π is defined by a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ as follows:

Init: $\mathcal{C}$ runs $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and gives mpk to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ is allowed to make the following three types of queries to $\mathcal{C}$:

Key extraction query: $\mathcal{A}$ is allowed to make the query on $y \in \mathcal{Y}$ to $\mathcal{C}$. Upon the query, $\mathcal{C}$ runs $\mathsf{sk}_y \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, y)$ and returns sk_y to $\mathcal{A}$.

Decryption query: $\mathcal{A}$ is allowed to make the query on (ct_x, y) to $\mathcal{C}$. Upon the query, $\mathcal{C}$ runs $\mathsf{sk}_y \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, y)$ and $\mathsf{M} \leftarrow \mathsf{Dec}(\mathsf{mpk}, \mathsf{ct}_x, \mathsf{sk}_y)$, and returns M to $\mathcal{A}$.

Trapdoor query: $\mathcal{A}$ is allowed to make the query on $y \in \mathcal{Y}$ to $\mathcal{C}$. Upon the query, $\mathcal{C}$ runs $\mathsf{sk}_y \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, y)$ and $\mathsf{td}_y \leftarrow \mathsf{Trapdoor}(\mathsf{mpk}, \mathsf{sk}_y)$, and returns td_y to $\mathcal{C}$.

Challenge query: $\mathcal{A}$ is allowed to make the query only once. Upon $\mathcal{A}$'s query on $x^* \in \mathcal{X}$, x^* should not satisfy the condition $\mathsf{P}(x^*, y) = 1$ for all the attributes $y \in \mathcal{Y}$ queried on key extraction queries in Phase 1. Then, $\mathcal{C}$ chooses $\mathsf{M}^* \leftarrow_\$ \mathcal{M}$ and runs $\mathsf{ct}^*_{x^*} \leftarrow \mathsf{Enc}(\mathsf{mpk}, x^*, \mathsf{M}^*)$. Finally, $\mathcal{C}$ returns $\mathsf{ct}^*_{x^*}$ to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ is allowed to make key extraction queries, decryption queries and trapdoor queries as in Phase 1 with the following exceptions:

Key extraction query: Upon $\mathcal{A}$'s query on $y \in \mathcal{Y}$, y should not satisfy the condition $\mathsf{P}(x^*, y) = 1$.

Decryption query: Upon $\mathcal{A}$'s query on (ct_x, y), $\mathsf{ct}_x = \mathsf{ct}^*_{x^*}$ does not hold.

Guess: At the end of the game, $\mathcal{A}$ returns $\widehat{\mathsf{M}} \in \mathcal{M}$ as a guess of M^*.

The adversary $\mathcal{A}$ wins in the above game if $\widehat{\mathsf{M}} = \mathsf{M}^*$ and the advantage is defined to

$$\mathsf{Adv}^{\text{OW-CCA2}}_{\Pi,\mathcal{A}}(\lambda) := \left| \Pr\left[\widehat{\mathsf{M}} = \mathsf{M}^*\right] - \frac{1}{|\mathcal{M}|} \right|.$$

If $\mathsf{Adv}^{\text{OW-CCA2}}_{\Pi,\mathcal{A}}(\lambda)$ is negligible in the security parameter λ for all PPT adversaries $\mathcal{A}$, an ABEET scheme Π is said to satisfy adaptive OW-CCA2 security against Type-I adversaries.

Definition 2.6 (Adaptive IND-CCA2 Security against Type-II Adversaries)**.** The adaptive IND-CCA2 security against Type-II adversaries of an ABEET scheme Π is defined by a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ as follows:

Init: $\mathcal{C}$ runs $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and gives mpk to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ is allowed to make the following three types of queries to $\mathcal{C}$:

Key extraction query: $\mathcal{A}$ is allowed to make the query on $y \in \mathcal{Y}$ to $\mathcal{C}$. Upon the query, $\mathcal{C}$ runs $\mathsf{sk}_y \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, y)$ and returns sk_y to $\mathcal{A}$.

Decryption query: $\mathcal{A}$ is allowed to make the query on (ct_x, y) to $\mathcal{C}$. Upon the query, $\mathcal{C}$ runs $\mathsf{sk}_y \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, y)$ and $\mathsf{M} \leftarrow \mathsf{Dec}(\mathsf{mpk}, \mathsf{ct}_x, \mathsf{sk}_y)$, and returns M to $\mathcal{A}$.

Trapdoor query: $\mathcal{A}$ is allowed to make the query on $y \in \mathcal{Y}$ to $\mathcal{C}$. Upon the query, $\mathcal{C}$ runs $\mathsf{sk}_y \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, y)$ and $\mathsf{td}_y \leftarrow \mathsf{Trapdoor}(\mathsf{mpk}, \mathsf{sk}_y)$, and returns td_y to $\mathcal{C}$.

Challenge query: $\mathcal{A}$ is allowed to make the query only once. Upon $\mathcal{A}$'s query on $(x^*, \mathsf{M}^*_0, \mathsf{M}^*_1) \in \mathcal{X} \times \mathcal{M}^2$, $|\mathsf{M}^*_0| = |\mathsf{M}^*_1|$ holds and x^* should not satisfy the condition $\mathsf{P}(x^*, y) = 1$ for all the attributes $y \in \mathcal{Y}$ queried on key extraction queries and trapdoor queries in Phase 1. Then, $\mathcal{C}$ flips a coin $\mathsf{coin} \leftarrow_\$ \{0, 1\}$ and runs $\mathsf{ct}^*_{x^*} \leftarrow \mathsf{Enc}(\mathsf{mpk}, x^*, \mathsf{M}^*_{\mathsf{coin}})$. Finally, $\mathcal{C}$ returns $\mathsf{ct}^*_{x^*}$ to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ is allowed to make key extraction queries, decryption queries and trapdoor queries as in Phase 1 with the following exceptions:

Key extraction query: Upon $\mathcal{A}$'s query on $y \in \mathcal{Y}$, y should not satisfy the condition $\mathsf{P}(x^*, y) = 1$.

Decryption query: Upon $\mathcal{A}$'s query on (ct_x, y), $\mathsf{ct}_x = \mathsf{ct}^*_{x^*}$ does not hold.

Trapdoor query: Upon $\mathcal{A}$'s query on $y \in \mathcal{Y}$, y should not satisfy the condition $\mathsf{P}(x^*, y) = 1$.

Guess: At the end of the game, $\mathcal{A}$ outputs $\widehat{\mathsf{coin}} \in \{0, 1\}$ as a guess of coin.

The adversary $\mathcal{A}$ wins in the above game if $\widehat{\mathsf{coin}} = \mathsf{coin}$ and the advantage is defined to

$$\mathsf{Adv}^{\text{IND-CCA2}}_{\Pi,\mathcal{A}}(\lambda) := \left| \Pr\left[\widehat{\mathsf{coin}} = \mathsf{coin}\right] - \frac{1}{2} \right|.$$

If $\mathsf{Adv}^{\text{IND-CCA2}}_{\Pi,\mathcal{A}}(\lambda)$ is negligible in the security parameter λ for all PPT adversaries $\mathcal{A}$, an ABEET scheme Π is said to satisfy adaptive IND-CCA2 security against Type-II adversaries.

Remark 3. As the case of ABE, we define selective security and semi-adaptive security for ABEET by following Remark 1.

3 Proposed Generic Construction

In this section, we provide a generic construction of ABEET by following the discussion in Section 1.3. In section 3.1, we show the construction. In Section 3.2, we prove the correctness of our construction.

3.1 Our construction

In this section, we construct an ABEET scheme Π for a predicate P from an ABE scheme Π_{ABE}, an OTS scheme Γ and a hash function H. Here, we assume that plaintext spaces $\mathcal{M}$ of ABE and ABEET are the same. Moreover, $\mathcal{M}$ is the same as the domain of the hash function H and the range of $\mathcal{R}$ is a subset of $\mathcal{M}$.

$\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$**:** Run

- $(\mathsf{ABE.mpk}, \mathsf{ABE.msk}) \leftarrow \mathsf{ABE.Setup}(1^\lambda)$,

and output $\mathsf{mpk} := (\mathsf{ABE.mpk}, \Gamma, \mathsf{H})$ and $\mathsf{msk} := \mathsf{ABE.msk}$.

$\mathsf{Enc}(\mathsf{mpk}, x, \mathsf{M}) \to \mathsf{ct}_x$**:** Parse $\mathsf{mpk} = (\mathsf{ABE.mpk}, \Gamma, \mathsf{H})$. Run

- $(\mathsf{verk}, \mathsf{sigk}) \leftarrow \mathsf{Sig.Setup}(1^\lambda)$,
- $\mathsf{ABE.ct}_{x,0,\mathsf{verk}} \leftarrow \mathsf{ABE.Enc}(\mathsf{ABE.mpk}, (x, 0, \mathsf{verk}), \mathsf{M})$,
- $\mathsf{ABE.ct}_{x,1,\mathsf{verk}} \leftarrow \mathsf{ABE.Enc}(\mathsf{ABE.mpk}, (x, 1, \mathsf{verk}), \mathsf{H}(\mathsf{M}))$,
- $\sigma \leftarrow \mathsf{Sig.Sign}(\mathsf{sigk}, [\mathsf{ABE.ct}_{x,0,\mathsf{verk}} \| \mathsf{ABE.ct}_{x,1,\mathsf{verk}}])$.

Output $\mathsf{ct}_x = (\mathsf{verk}, \mathsf{ABE.ct}_{x,0,\mathsf{verk}}, \mathsf{ABE.ct}_{x,1,\mathsf{verk}}, \sigma)$.

$\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, y) \to \mathsf{sk}_y$**:** Parse $\mathsf{mpk} = (\mathsf{ABE.mpk}, \Gamma, \mathsf{H})$ and $\mathsf{msk} = \mathsf{ABE.msk}$. Run

- $\mathsf{ABE.sk}_y \leftarrow \mathsf{ABE.KeyGen}(\mathsf{ABE.mpk}, \mathsf{ABE.msk}, y)$.

Output $\mathsf{sk}_y := \mathsf{ABE.sk}_y$.

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{ct}_x, \mathsf{sk}_y) \to \mathsf{M}$ or $\perp$**:** Parse $\mathsf{mpk} = (\mathsf{ABE.mpk}, \Gamma, \mathsf{H})$, $\mathsf{ct}_x = (\mathsf{verk}, \mathsf{ABE.ct}_{x,0,\mathsf{verk}}, \mathsf{ABE.ct}_{x,1,\mathsf{verk}}, \sigma)$, and $\mathsf{sk}_y = \mathsf{ABE.sk}_y$. If it holds that

- $0 \leftarrow \mathsf{Sig.Vrfy}(\mathsf{verk}, \left[\mathsf{ABE.ct}_{x,0,\mathsf{verk}} \| \mathsf{ABE.ct}_{x,1,\mathsf{verk}}\right], \sigma) \vee \mathsf{P}(x, y) = 0$,

output $\perp$. Otherwise, run

- $\mathsf{ABE.sk}_{y,0,\mathsf{verk}} \leftarrow \mathsf{ABE.Delegate}(\mathsf{ABE.mpk}, \mathsf{ABE.sk}_y, (y, 0, \mathsf{verk}))$,
- $\mathsf{ABE.sk}_{y,1,\mathsf{verk}} \leftarrow \mathsf{ABE.Delegate}(\mathsf{ABE.mpk}, \mathsf{ABE.sk}_y, (y, 1, \mathsf{verk}))$,
- $\mathsf{M} \leftarrow \mathsf{ABE.Dec}(\mathsf{ABE.mpk}, \mathsf{ABE.ct}_{x,0,\mathsf{verk}}, \mathsf{ABE.sk}_{y,0,\mathsf{verk}})$,
- $h \leftarrow \mathsf{ABE.Dec}(\mathsf{ABE.mpk}, \mathsf{ABE.ct}_{x,1,\mathsf{verk}}, \mathsf{ABE.sk}_{y,1,\mathsf{verk}})$.

Output M if $\mathsf{H}(\mathsf{M}) = h$ holds and $\perp$ otherwise.

$\mathsf{Trapdoor}(\mathsf{mpk}, \mathsf{sk}_y) \to \mathsf{td}_y$**:** Parse $\mathsf{mpk} = (\mathsf{ABE.mpk}, \Gamma, \mathsf{H})$ and $\mathsf{sk}_y = \mathsf{ABE.sk}_y$. Run

- $\mathsf{ABE.sk}_{y,1} \leftarrow \mathsf{ABE.Delegate}(\mathsf{ABE.mpk}, \mathsf{ABE.sk}_y, (y, 1))$.

Output $\mathsf{td}_y := \mathsf{ABE.sk}_{y,1}$.

$\mathsf{Test}(\mathsf{mpk}, \mathsf{ct}_x, \mathsf{td}_y, \mathsf{ct}_{x'}, \mathsf{td}_{y'}) \to 1 \text{ or } 0$**:** Parse $\mathsf{mpk} = (\mathsf{ABE.mpk}, \Gamma, \mathsf{H})$, $\mathsf{ct}_x = (\mathsf{verk}, \mathsf{ABE.ct}_{x,0,\mathsf{verk}}, \mathsf{ABE.ct}_{x,1,\mathsf{verk}}, \sigma)$, $\mathsf{ct}_{x'} = (\mathsf{verk}', \mathsf{ABE.ct}_{x',0,\mathsf{verk}'}, \mathsf{ABE.ct}_{x',1,\mathsf{verk}'}, \sigma')$, $\mathsf{td}_y = \mathsf{ABE.sk}_{y,1}$, and $\mathsf{td}_{y'} = \mathsf{ABE.sk}_{y',1}$. If it holds that

- $0 \leftarrow \mathsf{Sig.Vrfy}(\mathsf{verk}, [\mathsf{ABE.ct}_{x,0,\mathsf{verk}} \| \mathsf{ABE.ct}_{x,1,\mathsf{verk}}], \sigma) \ \vee 0 \leftarrow \mathsf{Sig.Vrfy}(\mathsf{verk}', [\mathsf{ABE.ct}_{x',0,\mathsf{verk}'} \| \mathsf{ABE.ct}_{x',1,\mathsf{verk}'}], \sigma')$,

output 0. Otherwise, run

- $\mathsf{ABE.sk}_{y,1,\mathsf{verk}} \leftarrow \mathsf{ABE.Delegate}(\mathsf{ABE.mpk}, \mathsf{ABE.sk}_{y,1}, (y, 1, \mathsf{verk}))$,
- $\mathsf{ABE.sk}_{y',1,\mathsf{verk}'} \leftarrow \mathsf{ABE.Delegate}(\mathsf{ABE.mpk}, \mathsf{ABE.sk}_{y',1}, (y', 1, \mathsf{verk}'))$,
- $h \leftarrow \mathsf{ABE.Dec}(\mathsf{ABE.mpk}, \mathsf{ABE.ct}_{x,1,\mathsf{verk}}, \mathsf{ABE.sk}_{y,1,\mathsf{verk}})$,
- $h' \leftarrow \mathsf{ABE.Dec}(\mathsf{ABE.mpk}, \mathsf{ABE.ct}_{x',1,\mathsf{verk}'}, \mathsf{ABE.sk}_{y',1,\mathsf{verk}'})$.

Output 1 if $h = h'$ and 0 otherwise.

3.2 Correctness

We prove the correctness of our ABEET construction as follows.

Theorem 3.1. Our ABEET scheme Π satisfies correctness if the underlying ABE scheme Π_{ABE} and OTS scheme Γ satisfy correctness, and the hash function H satisfies collision resistance.

Proof. We can prove the condition (1) by using the correctness of the underlying ABE scheme Π_{ABE} and the underlying OTS scheme Γ. For all $\lambda \in \mathbb{N}$, all $(\mathsf{ABE.mpk}, \mathsf{ABE.msk}) \leftarrow \mathsf{ABE.Setup}(1^\lambda)$ and Γ, all $\mathsf{M} \in \mathcal{M}$, all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ such that $\mathsf{P}(x, y) = 1$, it is required that

$$\mathsf{Sig.Vrfy}(\mathsf{verk}, [\mathsf{ABE.ct}_{x,0,\mathsf{verk}} \| \mathsf{ABE.ct}_{x,1,\mathsf{verk}}], \sigma) \to 1 \wedge \mathsf{M}' = \mathsf{M} \wedge h = \mathsf{H}(\mathsf{M})$$

holds with overwhelming probability, where

- $(\mathsf{verk}, \mathsf{sigk}) \leftarrow \mathsf{Sig.Setup}(1^\lambda)$,
- $\mathsf{ABE.ct}_{x,0,\mathsf{verk}} \leftarrow \mathsf{ABE.Enc}(\mathsf{ABE.mpk}, (x, 0, \mathsf{verk}), \mathsf{M})$,
- $\mathsf{ABE.ct}_{x,1,\mathsf{verk}} \leftarrow \mathsf{ABE.Enc}(\mathsf{ABE.mpk}, (x, 1, \mathsf{verk}), \mathsf{H}(\mathsf{M}))$,
- $\sigma \leftarrow \mathsf{Sig.Sign}(\mathsf{sigk}, [\mathsf{ct}_{x,0,\mathsf{verk}} \| \mathsf{ct}_{x,1,\mathsf{verk}}])$,
- $\mathsf{ABE.sk}_y \leftarrow \mathsf{ABE.KeyGen}(\mathsf{ABE.mpk}, \mathsf{ABE.msk}, y)$,
- $\mathsf{ABE.sk}_{y,0,\mathsf{verk}} \leftarrow \mathsf{ABE.Delegate}(\mathsf{ABE.mpk}, \mathsf{ABE.sk}_y, (y, 0, \mathsf{verk}))$,
- $\mathsf{ABE.sk}_{y,1,\mathsf{verk}} \leftarrow \mathsf{ABE.Delegate}(\mathsf{ABE.mpk}, \mathsf{ABE.sk}_y, (y, 1, \mathsf{verk}))$,
- $\mathsf{M}' \leftarrow \mathsf{ABE.Dec}(\mathsf{ABE.mpk}, \mathsf{ABE.ct}_{x,0,\mathsf{verk}}, \mathsf{ABE.sk}_{y,0,\mathsf{verk}})$,
- $h \leftarrow \mathsf{ABE.Dec}(\mathsf{ABE.mpk}, \mathsf{ABE.ct}_{x,1,\mathsf{verk}}, \mathsf{ABE.sk}_{y,1,\mathsf{verk}})$.

The correctness of the OTS scheme Γ ensures that $\mathsf{Sig.Vrfy}(\mathsf{verk}, [\mathsf{ABE.ct}_{x,0,\mathsf{verk}} \| \mathsf{ABE.ct}_{x,1,\mathsf{verk}}], \sigma) \rightarrow 1$ holds with overwhelming probability. Moreover, the correctness of the ABE scheme Π_{ABE} ensures that $\mathsf{M} = \mathsf{M}' \wedge h = \mathsf{H}(\mathsf{M})$ holds with overwhelming probability. Therefore, the condition (1) holds.

We can prove the condition (2) by using the correctness of the underlying ABE scheme Π_{ABE} and the underlying OTS scheme Γ. For all $\lambda \in \mathbb{N}$, all $(\mathsf{ABE.mpk}, \mathsf{ABE.msk}) \leftarrow \mathsf{ABE.Setup}(1^\lambda)$ and Γ, all $\mathsf{M} \in \mathcal{M}$, all $(x_0, x_1, y_0, y_1) \in \mathcal{X}^2 \times \mathcal{Y}^2$ such that $\wedge_{i \in \{0,1\}} \mathsf{P}(x_i, y_i) = 1$, it is required that

$$(\wedge_{i \in \{0,1\}} \mathsf{Sig.Vrfy}(\mathsf{verk}_i, \left[\mathsf{ABE.ct}_{x_i,0,\mathsf{verk}_i} \| \mathsf{ABE.ct}_{x_i,1,\mathsf{verk}_i}\right], \sigma_i) \rightarrow 1) \wedge h_0 = h_1$$

holds with overwhelming probability, where for $i \in \{0,1\}$

- $(\mathsf{verk}_i, \mathsf{sigk}_i) \leftarrow \mathsf{Sig.Setup}(1^\lambda)$,
- $\mathsf{ABE.ct}_{x_i,0,\mathsf{verk}} \leftarrow \mathsf{ABE.Enc}(\mathsf{ABE.mpk}, (x_i, 0, \mathsf{verk}_i), \mathsf{M})$,
- $\mathsf{ABE.ct}_{x_i,1,\mathsf{verk}} \leftarrow \mathsf{ABE.Enc}(\mathsf{ABE.mpk}, (x_i, 1, \mathsf{verk}_i), \mathsf{H}(\mathsf{M}))$,
- $\sigma_i \leftarrow \mathsf{Sig.Sign}(\mathsf{sigk}_i, \left[\mathsf{ct}_{x_i,0,\mathsf{verk}_i} \| \mathsf{ct}_{x_i,1,\mathsf{verk}_i}\right])$,
- $\mathsf{ABE.sk}_{y_i} \leftarrow \mathsf{ABE.KeyGen}(\mathsf{ABE.mpk}, \mathsf{ABE.msk}, y_i)$,
- $\mathsf{ABE.sk}_{y_i,1,\mathsf{verk}_i} \leftarrow \mathsf{ABE.Delegate}(\mathsf{ABE.mpk}, \mathsf{ABE.sk}_{y_i}, (y_i, 1, \mathsf{verk}_i))$,
- $h_i \leftarrow \mathsf{ABE.Dec}(\mathsf{ABE.mpk}, \mathsf{ABE.ct}_{x_i,1,\mathsf{verk}_i}, \mathsf{ABE.sk}_{y_i,1,\mathsf{verk}_i})$.

The correctness of the OTS scheme Γ ensures that $\mathsf{Sig.Vrfy}(\mathsf{verk}_i, [\mathsf{ABE.ct}_{x_i,0,\mathsf{verk}_i} \| \mathsf{ABE.ct}_{x_i,1,\mathsf{verk}_i}], \sigma_i) \rightarrow 1$ holds with overwhelming probability. Moreover, the correctness of the ABE scheme Π_{ABE} ensures that $h_i = \mathsf{H}(\mathsf{M})$ for $i \in \{0,1\}$ holds with overwhelming probability. Therefore, the condition (2) holds.

We can prove the condition (3) by using the correctness of the underlying ABE scheme Π_{ABE} and collision resistance of underlying hash function H. For this purpose, we use an adversary $\mathcal{A}$ for breaking the condition (3) to construct a PPT adversary $\mathcal{B}$ that breaks the collision resistance of H. Here, we say that $\mathcal{A}$ breaks the condition (3) if it holds that $\mathsf{M}_0 \neq \mathsf{M}_1 \wedge \mathsf{Test}(\mathsf{mpk}, \mathsf{ct}_{x_0}, \mathsf{td}_{y_0}, \mathsf{ct}_{x_1}, \mathsf{td}_{y_1}) \rightarrow 1$, where $(\mathsf{M}_0, \mathsf{M}_1) \leftarrow \mathcal{A}(\mathsf{mpk}, \mathsf{msk}), \mathsf{ct}_{x_i} \leftarrow \mathsf{Enc}(\mathsf{mpk}, x_i, \mathsf{M})$, $\mathsf{sk}_{y_i} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, y_i)$ and $\mathsf{td}_{y_i} \leftarrow \mathsf{Trapdoor}(\mathsf{mpk}, \mathsf{sk}_{y_i})$ for $i = 0, 1$. For all $\lambda \in \mathbb{N}$, all $(\mathsf{ABE.mpk}, \mathsf{ABE.msk}) \leftarrow \mathsf{ABE.Setup}(1^\lambda)$ and (Γ, H), all PPT adversaries $\mathcal{A}$, all $(x_0, x_1, y_0, y_1) \in \mathcal{X}^2 \times \mathcal{Y}^2$ such that $\wedge_{i \in \{0,1\}} \mathsf{P}(x_i, y_i) = 1$, after $\mathcal{A}$ outputs $(\mathsf{M}_0, \mathsf{M}_1)$, $\mathcal{B}$ also outputs the same $(\mathsf{M}_0, \mathsf{M}_1)$. If $\mathcal{A}$ breaks the condition (3), it holds that $\mathsf{M}_0 \neq \mathsf{M}_1 \wedge h_0 = h_1$, where for $i \in \{0,1\}$

- $(\mathsf{verk}_i, \mathsf{sigk}_i) \leftarrow \mathsf{Sig.Setup}(1^\lambda)$,
- $\mathsf{ABE.ct}_{x_i,1,\mathsf{verk}_i} \leftarrow \mathsf{ABE.Enc}(\mathsf{ABE.mpk}, (x_i, 1, \mathsf{verk}_i), \mathsf{H}(\mathsf{M}_i))$,
- $\mathsf{ABE.sk}_{y_i} \leftarrow \mathsf{ABE.KeyGen}(\mathsf{ABE.mpk}, \mathsf{ABE.msk}, y_i)$,
- $\mathsf{ABE.sk}_{y_i,1,\mathsf{verk}_i} \leftarrow \mathsf{ABE.Delegate}(\mathsf{ABE.mpk}, \mathsf{ABE.sk}_{y_i}, (y_i, 1, \mathsf{verk}_i))$,
- $h_i \leftarrow \mathsf{ABE.Dec}(\mathsf{ABE.mpk}, \mathsf{ABE.ct}_{x_i,1,\mathsf{verk}_i}, \mathsf{ABE.sk}_{y_i,1,\mathsf{verk}_i})$.

The correctness of the ABE scheme Π_{ABE} ensures that $h_i = \mathsf{H}(\mathsf{M}_i)$ hold for $i \in \{0,1\}$ with overwhelming probability. Therefore, if $\mathcal{A}$ breaks the condition (3), $\mathcal{B}$ breaks the collision resistance of H with overwhelming probability since it holds that $\mathsf{M}_0 \neq \mathsf{M}_1 \wedge \mathsf{H}(\mathsf{M}_0) = \mathsf{H}(\mathsf{M}_1)$. Therefore, the condition (3) holds.

From the above, it is proved that our proposed construction is correct. □

4 Security

In this section, we provide security proofs of our generic construction given in Section 3.1. Specifically, we prove OW-CCA2 security against Type-I adversaries and IND-CCA2 security against Type-II adversaries in Sections 4.1 and 4.2, respectively.

4.1 OW-CCA2 Security against Type-I Adversaries

Theorem 4.1 (OW-CCA2 Security against Type-I Adversaries)**.** If the underlying ABE scheme Π_{ABE} satisfies adaptive (resp. semi-adaptive, selective) IND-CPA security, OTS scheme Γ satisfies strong unforgeability, and H satisfies one-wayness, then our proposed ABEET scheme Π satisfies adaptive (resp. semi-adaptive, selective) OW-CCA2 security against Type-I adversaries.

Proof. Here, we prove Theorem 4.1 as the case of *adaptive* security. We note that the proofs for semi-adaptive security and selective security are essentially the same.

Let $\mathsf{ct}^*_{x^*} = (\mathsf{verk}^*, \mathsf{ABE.ct}^*_{x^*,0,\mathsf{verk}^*}, \mathsf{ABE.ct}^*_{x^*,1,\mathsf{verk}^*}, \sigma^*)$ be the challenge ciphertext for the target attribute x^*. We prove the theorem via game sequence $\mathbf{Game}_0$, $\mathbf{Game}_1$, and $\mathbf{Game}_2$. Let W_i denote an event that $\mathcal{A}$ wins in $\mathbf{Game}_i$ for $i \in \{0,1,2\}$.

$\mathbf{Game}_0$: This game is the same as the original adaptive OW-CCA2 security game in Definition 2.5 between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$.

$\mathbf{Game}_1$: This game is the same as $\mathbf{Game}_0$ except that if $\mathcal{A}$ makes the decryption queries on $(\mathsf{ct}_x, y) = ((\mathsf{verk}, \mathsf{ABE.ct}_{x,0,\mathsf{verk}}, \mathsf{ABE.ct}_{x,1,\mathsf{verk}}, \sigma), y)$ such that

$$\begin{aligned}\mathsf{verk} = \mathsf{verk}^* \wedge \mathsf{Sig.Vrfy}(\mathsf{verk}, \left[\mathsf{ABE.ct}_{x,0,\mathsf{verk}} \| \mathsf{ABE.ct}_{x,1,\mathsf{verk}}\right], \sigma) \to 1 \\ \wedge (\mathsf{ABE.ct}_{x,0,\mathsf{verk}}, \mathsf{ABE.ct}_{x,1,\mathsf{verk}}, \sigma) \neq (\mathsf{ABE.ct}^*_{x^*,0,\mathsf{verk}^*}, \mathsf{ABE.ct}^*_{x^*,1,\mathsf{verk}^*}, \sigma^*)\end{aligned}$$

then $\mathcal{C}$ aborts the game and returns $\mathsf{M} \leftarrow_{\$} \mathcal{M}$. Let E denote an event that $\mathcal{A}$ makes such decryption queries.

We show that $\mathbf{Game}_0$ and $\mathbf{Game}_1$ are computationally indistinguishable from $\mathcal{A}$'s view if the OTS scheme Γ satisfies strong unforgeability. For this purpose, we use $\mathcal{A}$ to construct a PPT adversary $\mathcal{F}$ that breaks strong unforgeability of Γ. Let $\mathsf{OTS}.\mathcal{C}$ denote a challenger of the strong unforgeability game of Γ. $\mathsf{OTS}.\mathcal{C}$ begins the strong unforgeability game and gives verk^* to $\mathcal{F}$. Then, $\mathcal{F}$ begins the OW-CCA2 security game with $\mathcal{A}$ by running $(\mathsf{ABE.mpk}, \mathsf{ABE.msk}) \leftarrow \mathsf{ABE.Setup}(1^\lambda)$ and giving $\mathsf{mpk} = (\mathsf{ABE.mpk}, \Gamma, \mathsf{H})$ to $\mathcal{A}$. Since $\mathcal{F}$ obtains $\mathsf{msk} = \mathsf{ABE.msk}$, it can answer $\mathcal{A}$'s key extraction queries and trapdoor queries. Similarly, if E does not happen, $\mathcal{F}$ can answer $\mathcal{A}$'s decryption queries. In contrast, if E happens, $\mathcal{F}$ aborts the OW-CCA2 security game and returns $\mathsf{M} \leftarrow_{\$} \mathcal{M}$. Moreover, $\mathcal{F}$ returns $(\left[\mathsf{ABE.ct}_{x,0,\mathsf{verk}} \| \mathsf{ABE.ct}_{x,1,\mathsf{verk}}\right], \sigma)$ to $\mathsf{OTS}.\mathcal{C}$ as a pair of a message and a forged signature. Upon $\mathcal{A}$'s challenge query on x^*, $\mathcal{F}$ chooses $\mathsf{M}^* \leftarrow_{\$} \mathcal{M}$ and runs $\mathsf{ABE.ct}^*_{x^*,0,\mathsf{verk}^*} \leftarrow \mathsf{ABE.Enc}(\mathsf{ABE.mpk}, (x^*, 0, \mathsf{verk}^*), \mathsf{M}^*)$ and $\mathsf{ABE.ct}^*_{x^*,1,\mathsf{verk}^*} \leftarrow \mathsf{ABE.Enc}(\mathsf{ABE.mpk}, (x^*, 1, \mathsf{verk}^*), \mathsf{H}(\mathsf{M}^*))$. Then, $\mathcal{F}$ makes a query on $[\mathsf{ABE.ct}^*_{x^*,0,\mathsf{verk}^*} \| \mathsf{ABE.ct}^*_{x^*,1,\mathsf{verk}^*}]$ to $\mathsf{OTS}.\mathcal{C}$ and receives σ^*. $\mathcal{F}$ gives $\mathsf{ct}^*_{x^*} = (\mathsf{verk}^*, \mathsf{ABE.ct}^*_{x^*,0,\mathsf{verk}^*}, \mathsf{ABE.ct}^*_{x^*,1,\mathsf{verk}^*}, \sigma^*)$ to $\mathcal{A}$.

Observe that all $\mathcal{F}$'s behavior except the challenge query does not depend on verk^* if E does not occur. Thus, $\mathcal{F}$ perfectly simulates $\mathbf{Game}_0$ if E does not happen. Similarly, $\mathcal{F}$ perfectly simulates $\mathbf{Game}_1$ if E happens. In this case, $\mathcal{F}$ successfully breaks the strong unforgeability of Γ. Therefore, we have

$$\Pr[E] \leq \mathsf{Adv}^{\mathsf{OTS}}_{\Gamma,\mathcal{F}}(\lambda).$$

If E happens in $\mathbf{Game}_1$, $\mathcal{F}$ outputs a random $\mathsf{M} \leftarrow_{\$} \mathcal{M}$. In other words, it holds that $\Pr[W_1 \mid E] = 1/|\mathcal{M}|$. Therefore, we have

$$\Pr[W_1] = \Pr[W_1 \mid E] \Pr[E] + \Pr[W_1 \mid \neg E] \Pr[\neg E]$$

$$= \frac{1}{|\mathcal{M}|} \cdot \Pr[E] + \Pr[W_1 \mid \neg E] \Pr[\neg E].$$

If E does not happen, $\mathbf{Game}_0$ and $\mathbf{Game}_1$ are the same from $\mathcal{A}$'s view. In other words, it holds that

$$\Pr[W_1 \mid \neg E] \Pr[\neg E] = \Pr[W_0](1 - \Pr[E]).$$

Therefore, we have

$$\begin{aligned}\Pr[W_1] &= \frac{1}{|\mathcal{M}|} \cdot \Pr[E] + \Pr[W_0] - \Pr[W_0] \cdot \Pr[E] \\ &= \Pr[W_0] + \left(\frac{1}{|\mathcal{M}|} - \Pr[W_0] \right) \cdot \Pr[E] \\ &\geq \Pr[W_0] - \Pr[E].\end{aligned}$$

Therefore, we have

$$|\Pr[W_0] - \Pr[W_1]| \leq \Pr[E] \leq \mathsf{Adv}^{\mathsf{OTS}}_{\Gamma,\mathcal{F}}(\lambda). \tag{1}$$

Next, we define the $\mathbf{Game}_2$ as follows.

$\mathbf{Game}_2$: This game is the same as $\mathbf{Game}_1$ except the way $\mathcal{C}$ creates the challenge ciphertext $\mathsf{ct}^*_{x^*} = (\mathsf{verk}^*, \mathsf{ABE.ct}^*_{x^*,0,\mathsf{verk}^*}, \mathsf{ABE.ct}^*_{x^*,1,\mathsf{verk}^*}, \sigma^*)$. In short, $\mathsf{ABE.ct}^*_{x^*,0,\mathsf{verk}^*}$ is an encryption of the challenge plaintext M^* in $\mathbf{Game}_1$. In contrast, $\mathsf{ABE.ct}^*_{x^*,0,\mathsf{verk}^*}$ is an encryption of a plaintext $\mathsf{M} \in \mathcal{M}$ in $\mathbf{Game}_2$, where a distribution of $\mathsf{M} \in \mathcal{M}$ is independent of M^* such as the uniform distribution over $\mathcal{M}$.

We show that $\mathbf{Game}_1$ and $\mathbf{Game}_2$ are computationally indistinguishable from $\mathcal{A}$'s view if the ABE scheme Π_{ABE} satisfies IND-CPA security. For this purpose, we use $\mathcal{A}$ to construct a PPT adversary $\mathcal{B}$ that breaks IND-CPA security of Π_{ABE}. Let $\mathsf{ABE}.\mathcal{C}$ denote a challenger of the IND-CPA security game of Π_{ABE}. $\mathcal{B}$ runs $(\mathsf{verk}^*, \mathsf{sigk}) \leftarrow \mathsf{Sig.Setup}(1^\lambda)$. $\mathsf{ABE}.\mathcal{C}$ begins the IND-CPA security game and gives $\mathsf{ABE.mpk}$ to $\mathcal{B}$.[2] Then, $\mathcal{B}$ begins the OW-CCA2 security game with $\mathcal{A}$ by giving $\mathsf{mpk} = (\mathsf{ABE.mpk}, \Gamma, \mathsf{H})$ to $\mathcal{A}$.

In the Phase 1, $\mathcal{B}$ can answer all three types of queries by interacting with $\mathsf{ABE}.\mathcal{C}$ as follows.

- **Key extraction query**: Upon $\mathcal{A}$'s query on y, $\mathcal{B}$ makes a key extraction query on y to $\mathsf{ABE}.\mathcal{C}$ and receives $\mathsf{ABE.sk}_y$. Then, $\mathcal{B}$ sends $\mathsf{ABE.sk}_y$ to $\mathcal{A}$.
- **Decryption query**: If E happens, $\mathcal{B}$ aborts the game and returns $\mathsf{M} \leftarrow_{\$} \mathcal{M}$. Otherwise, upon $\mathcal{A}$'s query on $(\mathsf{ct}_x = (\mathsf{verk}, \mathsf{ABE.ct}_{x,0,\mathsf{verk}}, \mathsf{ABE.ct}_{x,1,\mathsf{verk}}, \sigma), y)$, $\mathcal{B}$ returns $\perp$ if $0 \leftarrow \mathsf{Sig.Vrfy}(\mathsf{verk}, [\mathsf{ABE.ct}_{x,0,\mathsf{verk}} \| \mathsf{ABE.ct}_{x,1,\mathsf{verk}}], \sigma) \vee \mathsf{P}(x, y) = 0$. Otherwise, $\mathcal{B}$ makes the key extraction queries on $(y, 0, \mathsf{verk})$ and $(y, 1, \mathsf{verk})$ to $\mathsf{ABE}.\mathcal{C}$ and receives $\mathsf{ABE.sk}_{y,0,\mathsf{verk}}$ and $\mathsf{ABE.sk}_{y,1,\mathsf{verk}}$. $\mathcal{B}$ runs $\mathsf{M} \leftarrow \mathsf{ABE.Dec}(\mathsf{ABE.mpk}, \mathsf{ABE.ct}_{x,0,\mathsf{verk}}, \mathsf{ABE.sk}_{y,0,\mathsf{verk}})$ and $h \leftarrow \mathsf{ABE.Dec}(\mathsf{ABE.mpk}, \mathsf{ABE.ct}_{x,1,\mathsf{verk}}, \mathsf{ABE.sk}_{y,1,\mathsf{verk}})$. $\mathcal{B}$ returns M to $\mathcal{A}$ if $\mathsf{H}(\mathsf{M}) = h$ holds and $\perp$ otherwise.
- **Trapdoor query**: Upon $\mathcal{A}$'s query on y, $\mathcal{B}$ makes a key extraction query on $(y, 1)$ to $\mathsf{ABE}.\mathcal{C}$ and receives $\mathsf{ABE.sk}_{y,1}$. Then, $\mathcal{B}$ sends $\mathsf{td}_y = \mathsf{ABE.sk}_{y,1}$ to $\mathcal{A}$.

[2] To prove selective security, after receiving x^* from $\mathcal{A}$, $\mathcal{B}$ sends $(x^*, 0, \mathsf{verk}^*)$ to $\mathsf{ABE}.\mathcal{C}$ and $\mathsf{ABE}.\mathcal{C}$ begins the IND-CPA security game. Similarly, to prove semi-adaptive security, just after receiving x^* from $\mathcal{A}$, $\mathcal{B}$ sends $(x^*, 0, \mathsf{verk}^*)$ to $\mathsf{ABE}.\mathcal{C}$ before any queries in Phase 1.

Upon $\mathcal{A}$'s challenge query on x^*, $\mathcal{B}$ chooses $\mathsf{M}^*, \mathsf{M} \leftarrow_{\$} \mathcal{M}$, makes the challenge query on $((x^*, 0, \mathsf{verk}^*), \mathsf{M}^*, \mathsf{M})$ to $\mathsf{ABE}.\mathcal{C}$, and receives $\mathsf{ABE}.\mathsf{ct}^*_{x^*,0,\mathsf{verk}^*}$. Here, $\mathsf{ABE}.\mathsf{ct}^*_{x^*,0,\mathsf{verk}^*}$ are encryptions of M^* and M if $\mathsf{coin} = 0$ and $\mathsf{coin} = 1$, respectively. $\mathcal{B}$ runs $\mathsf{ABE}.\mathsf{ct}^*_{x^*,1,\mathsf{verk}^*} \leftarrow \mathsf{ABE}.\mathsf{Enc}(\mathsf{ABE}.\mathsf{mpk}, (x^*, 1, \mathsf{verk}^*), \mathsf{H}(\mathsf{M}^*))$ and $\sigma^* \leftarrow \mathsf{Sig}.\mathsf{Sign}(\mathsf{sigk}, [\mathsf{ABE}.\mathsf{ct}^*_{x^*,0,\mathsf{verk}^*} \| \mathsf{ABE}.\mathsf{ct}^*_{x^*,1,\mathsf{verk}^*}])$. $\mathcal{B}$ gives $\mathsf{ct}^*_{x^*} = (\mathsf{verk}^*, \mathsf{ABE}.\mathsf{ct}^*_{x^*,0,\mathsf{verk}^*}, \mathsf{ABE}.\mathsf{ct}^*_{x^*,1,\mathsf{verk}^*}, \sigma^*)$ to $\mathcal{A}$. In the Phase 2, $\mathcal{B}$ can answer all three types of queries essentially in the same way as in Phase 1. After $\mathcal{A}$ outputs $\widehat{\mathsf{M}}$ as a guess of M^*, $\mathcal{B}$ outputs $\widehat{\mathsf{coin}} = 0$ if $\widehat{\mathsf{M}} = \mathsf{M}^*$ and $\widehat{\mathsf{coin}} = 1$ otherwise as a guess of coin flipped by $\mathsf{ABE}.\mathcal{C}$.

If $\mathsf{ABE}.\mathsf{ct}^*_{x^*,0,\mathsf{verk}^*}$ which $\mathcal{B}$ received from $\mathsf{ABE}.\mathcal{C}$ are encryptions of M^* and M, the challenge ciphertext $\mathsf{ct}^*_{x^*}$ distribute as in $\mathbf{Game}_1$ and $\mathbf{Game}_2$, respectively. Observe that all $\mathcal{B}$'s key extraction queries to $\mathsf{ABE}.\mathcal{C}$ are valid, where the challenge ciphertext attribute of the IND-CPA security game for an ABE scheme Π_{ABE} is $(x^*, 0, \mathsf{verk}^*)$. All $\mathcal{B}$'s key extraction queries to answer $\mathcal{A}$'s key extraction queries are valid since $\mathsf{P}(x^*, y) = 0$ holds. All $\mathcal{B}$'s key extraction queries to answer $\mathcal{A}$'s decryption queries are valid since $\mathsf{verk} \neq \mathsf{verk}^*$ holds for the third hierarchy. All $\mathcal{B}$'s key extraction queries to answer $\mathcal{A}$'s trapdoor queries are valid since $1 \neq 0$ for the second hierarchy.

We analyze the quantity of $|\Pr[W_1] - \Pr[W_2]|$. By definition, $\Pr[\mathsf{coin} = 0] = \Pr[\mathsf{coin} = 1] = 1/2$ holds. As we mentioned above, $\mathcal{B}$ perfectly simulates $\mathbf{Game}_1$ and $\mathbf{Game}_2$ if $\mathsf{coin} = 0$ and $\mathsf{coin} = 1$, respectively; thus, $\Pr\left[\widehat{\mathsf{coin}} = 0 \mid \mathsf{coin} = 0\right] = \Pr[W_1]$ and $\Pr\left[\widehat{\mathsf{coin}} = 0 \mid \mathsf{coin} = 1\right] = \Pr[W_2]$ hold. Therefore, we have

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{ABE}}_{\Pi_{\mathsf{ABE}},\mathcal{B}}(\lambda) &= \left|\Pr\left[\widehat{\mathsf{coin}} = \mathsf{coin}\right] - \frac{1}{2}\right| \\
&= \left|\Pr\left[\widehat{\mathsf{coin}} = 0 \mid \mathsf{coin} = 0\right]\Pr[\mathsf{coin} = 0] + \Pr\left[\widehat{\mathsf{coin}} = 1 \mid \mathsf{coin} = 1\right]\Pr[\mathsf{coin} = 1] - \frac{1}{2}\right| \\
&= \frac{1}{2}\left|\Pr[W_1] - (1 - \Pr\left[\widehat{\mathsf{coin}} = 1 \mid \mathsf{coin} = 1\right])\right| \\
&= \frac{1}{2}\left|\Pr[W_1] - \Pr\left[\widehat{\mathsf{coin}} = 0 \mid \mathsf{coin} = 1\right]\right| \\
&= \frac{1}{2}|\Pr[W_1] - \Pr[W_2]|.
\end{aligned}$$

In other words, it holds that

$$|\Pr[W_1] - \Pr[W_2]| = 2\mathsf{Adv}^{\mathsf{ABE}}_{\Pi_{\mathsf{ABE}},\mathcal{B}}(\lambda). \tag{2}$$

Finally, we show that it is computationally infeasible for $\mathcal{A}$ to win in $\mathbf{Game}_2$ if the hash function H satisfies one-wayness. For this purpose, we use $\mathcal{A}$ to construct a PPT adversary $\mathcal{D}$ that breaks one-wayness of H. $\mathcal{D}$ interacts with $\mathcal{A}$ in the same way as $\mathcal{B}$ except the creation of the challenge ciphertext $\mathsf{ct}^*_{x^*}$. Upon $\mathcal{A}$'s challenge query on x^*, $\mathcal{D}$ receives h^* such that $\mathsf{M}^* \leftarrow_{\$} \mathcal{M}, h^* = \mathsf{H}(\mathsf{M}^*)$. $\mathcal{D}$ chooses $\mathsf{M} \leftarrow_{\$} \mathcal{M}$ and runs $\mathsf{ABE}.\mathsf{ct}^*_{x^*,0,\mathsf{verk}^*} \leftarrow \mathsf{ABE}.\mathsf{Enc}(\mathsf{ABE}.\mathsf{mpk}, (x^*, 0, \mathsf{verk}^*), \mathsf{M})$, $\mathsf{ABE}.\mathsf{ct}^*_{x^*,1,\mathsf{verk}^*} \leftarrow \mathsf{ABE}.\mathsf{Enc}(\mathsf{ABE}.\mathsf{mpk}, (x^*, 1, \mathsf{verk}^*), h^*)$, and $\sigma^* \leftarrow \mathsf{Sig}.\mathsf{Sign}(\mathsf{sigk}, [\mathsf{ABE}.\mathsf{ct}^*_{x^*,0,\mathsf{verk}^*} \| \mathsf{ABE}.\mathsf{ct}^*_{x^*,1,\mathsf{verk}^*}])$. $\mathcal{D}$ sets the challenge ciphertext $\mathsf{ct}^*_{x^*} = (\mathsf{verk}^*, \mathsf{ABE}.\mathsf{ct}^*_{x^*,0,\mathsf{verk}^*}, \mathsf{ABE}.\mathsf{ct}^*_{x^*,1,\mathsf{verk}^*}, \sigma^*)$. After $\mathcal{A}$ outputs $\widehat{\mathsf{M}}$ as a guess of M^*, $\mathcal{D}$ outputs $\widehat{\mathsf{M}}$ if $\mathsf{H}(\widehat{\mathsf{M}}) = h^*$ and $\widehat{\mathsf{M}} \leftarrow_{\$} \mathcal{M}$ otherwise.

$\mathcal{D}$ perfectly simulates $\mathbf{Game}_2$. If $\mathcal{A}$ wins in $\mathbf{Game}_2$, $\mathcal{D}$ always breaks the one-wayness of H. Therefore, we have

$$\left|\Pr[W_2] - \frac{1}{|\mathcal{M}|}\right| \leq \mathsf{Adv}^{\mathsf{OW}}_{\mathsf{H},\mathcal{D}}(\lambda). \tag{3}$$

From (1) – (3), we have

$$\left|\Pr[W_0] - \frac{1}{|\mathcal{M}|}\right| \leq |\Pr[W_0] - \Pr[W_1]| + |\Pr[W_1] - \Pr[W_2]| + \left|\Pr[W_2] - \frac{1}{|\mathcal{M}|}\right|$$

$$\leq \mathsf{Adv}^{\mathsf{OTS}}_{\Gamma,\mathcal{F}}(\lambda) + 2\mathsf{Adv}^{\mathsf{ABE}}_{\Pi_{\mathsf{ABE}},\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathsf{OW}}_{\mathsf{H},\mathcal{D}}(\lambda).$$

□

4.2 IND-CCA2 Security against Type-II Adversaries

Theorem 4.2 (IND-CCA2 Security against Type-II Adversaries)**.** If the underlying ABE scheme Π_{ABE} satisfies adaptive (resp. semi-adaptive, selective) IND-CPA security and OTS scheme Γ satisfies strong unforgeability, then our proposed ABEET scheme Π satisfies adaptive (resp. semi-adaptive, selective) IND-CCA2 security against Type-II adversaries.

Proof. Here, we prove Theorem 4.2 as the case of *adaptive* security. We note that the proofs for semi-adaptive security and selective security are essentially the same.

Let $\mathsf{ct}^*_{x^*} = (\mathsf{verk}^*, \mathsf{ABE.ct}^*_{x^*,0,\mathsf{verk}^*}, \mathsf{ABE.ct}^*_{x^*,1,\mathsf{verk}^*}, \sigma^*)$ be the challenge ciphertext for the target attribute x^*. We prove the theorem via game sequence $\mathbf{Game}_0$, $\mathbf{Game}_1$, and $\mathbf{Game}_2$. Let W_i denote an event that $\mathcal{A}$ wins in $\mathbf{Game}_i$ for $i \in \{0,1,2\}$.

$\mathbf{Game}_0$: This game is the same as the original adaptive IND-CCA2 security game in Definition 2.6 between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$.

$\mathbf{Game}_1$: This game is the same as $\mathbf{Game}_0$ except that if the event E (which was defined in $\mathbf{Game}_1$ in the proof of Theorem 4.1) happens, then the challenger $\mathcal{C}$ aborts the game and returns $\mathsf{coin}' \leftarrow_{\$} \{0,1\}$. $\mathbf{Game}_0$ and $\mathbf{Game}_1$ are computationally indistinguishable from $\mathcal{A}$'s view if the OTS scheme Γ satisfies strong unforgeability. In particular, there is a PPT adversary $\mathcal{F}$ such that

$$|\Pr[W_0] - \Pr[W_1]| \leq \Pr[E] \leq \mathsf{Adv}^{\mathsf{OTS}}_{\Gamma,\mathcal{F}}(\lambda) \tag{4}$$

by following essentially the same discussion as in (1).

Next, we define the $\mathbf{Game}_2$ as follows.

$\mathbf{Game}_2$: This game is the same as $\mathbf{Game}_1$ except the way $\mathcal{C}$ creates the challenge ciphertext $\mathsf{ct}^*_{x^*} = (\mathsf{verk}^*, \mathsf{ABE.ct}^*_{x^*,0,\mathsf{verk}^*}, \mathsf{ABE.ct}^*_{x^*,1,\mathsf{verk}^*}, \sigma^*)$. In short, $\mathsf{ABE.ct}^*_{x^*,0,\mathsf{verk}^*}$ is an encryption of $\mathsf{M}^*_{\mathsf{coin}}$ in $\mathbf{Game}_1$. In contrast, $\mathsf{ABE.ct}^*_{x^*,0,\mathsf{verk}^*}$ is an encryption of a plaintext $\mathsf{M} \in \mathcal{M}$ in $\mathbf{Game}_2$, where a distribution of $\mathsf{M} \in \mathcal{M}$ is independent of $\mathsf{M}^*_0, \mathsf{M}^*_1$ such as the uniform distribution over $\mathcal{M}$.

We show that $\mathbf{Game}_1$ and $\mathbf{Game}_2$ are computationally indistinguishable from $\mathcal{A}$'s view if the ABE scheme Π_{ABE} satisfies IND-CPA security. For this purpose, we use $\mathcal{A}$ to construct a PPT adversary $\mathcal{B}$ that breaks IND-CPA security of Π_{ABE}. $\mathcal{B}$ interacts with $\mathcal{A}$ in the same way as $\mathcal{B}$ in the proof of Theorem 4.1 except the creation of the challenge ciphertext $\mathsf{ct}^*_{x^*}$ and Guess phase. In this proof, upon $\mathcal{A}$'s challenge query on $(x^*, \mathsf{M}^*_0, \mathsf{M}^*_1)$, $\mathcal{B}$ chooses $\mathsf{coin} \leftarrow_{\$} \{0,1\}$ and $\mathsf{M} \leftarrow_{\$} \mathcal{M}$, makes the challenge query on $((x^*, 0, \mathsf{verk}^*), \mathsf{M}^*_{\mathsf{coin}}, \mathsf{M})$ to $\mathsf{ABE}.\mathcal{C}$, and receives $\mathsf{ABE.ct}^*_{x^*,0,\mathsf{verk}^*}$. Here, $\mathsf{ABE.ct}^*_{x^*,0,\mathsf{verk}^*}$ are encryptions of $\mathsf{M}^*_{\mathsf{coin}}$ and M if $\mathsf{coin}' = 0$ and $\mathsf{coin}' = 1$, respectively. $\mathcal{B}$ runs $\mathsf{ABE.ct}^*_{x^*,1,\mathsf{verk}^*} \leftarrow \mathsf{ABE.Enc}(\mathsf{ABE.mpk}, (x^*, 1, \mathsf{verk}^*), \mathsf{H}(\mathsf{M}^*_{\mathsf{coin}}))$ and $\sigma^* \leftarrow \mathsf{Sig.Sign}(\mathsf{sigk}, [\mathsf{ABE.ct}^*_{x^*,0,\mathsf{verk}^*} \| \mathsf{ABE.ct}^*_{x^*,1,\mathsf{verk}^*}])$. $\mathcal{B}$ gives $\mathsf{ct}^*_{x^*} = (\mathsf{verk}^*, \mathsf{ABE.ct}^*_{x^*,0,\mathsf{verk}^*}, \mathsf{ABE.ct}^*_{x^*,1,\mathsf{verk}^*}, \sigma^*)$ to $\mathcal{A}$. After $\mathcal{A}$ outputs $\widehat{\mathsf{coin}}$ as a guess of coin flipped by $\mathcal{B}$, $\mathcal{B}$ outputs $\widehat{\mathsf{coin}}' = 0$ if $\widehat{\mathsf{coin}} = \mathsf{coin}$ and $\widehat{\mathsf{coin}}' = 1$ otherwise as a guess of coin' flipped by $\mathsf{ABE}.\mathcal{C}$.

$\mathcal{B}$ perfectly simulates $\mathbf{Game}_1$ and $\mathbf{Game}_2$ if $\mathsf{coin}' = 0$ and $\mathsf{coin}' = 1$, respectively, by following essentially the same discussion as in the proof of Theorem 4.1. We analyze the quantity of $|\Pr[W_1] - \Pr[W_2]|$. In particular, we have

$$\mathsf{Adv}^{\mathsf{ABE}}_{\Pi_{\mathsf{ABE}},\mathcal{B}}(\lambda) = \left|\Pr\left[\widehat{\mathsf{coin}}' = \mathsf{coin}'\right] - \frac{1}{2}\right|$$

$$
\begin{aligned}
&= \left|\Pr\left[\widehat{\mathsf{coin}}' = 0 \mid \mathsf{coin}' = 0\right]\Pr[\mathsf{coin}' = 0] + \Pr\left[\widehat{\mathsf{coin}}' = 1 \mid \mathsf{coin}' = 1\right]\Pr[\mathsf{coin}' = 1] - \frac{1}{2}\right| \\
&= \frac{1}{2}\left|\Pr[W_1] - (1 - \Pr\left[\widehat{\mathsf{coin}}' = 1 \mid \mathsf{coin}' = 1\right])\right| \\
&= \frac{1}{2}\left|\Pr[W_1] - \Pr\left[\widehat{\mathsf{coin}}' = 0 \mid \mathsf{coin}' = 1\right]\right| \\
&= \frac{1}{2}|\Pr[W_1] - \Pr[W_2]|.
\end{aligned}
$$

In other words, it holds that

$$
|\Pr[W_1] - \Pr[W_2]| = 2\mathsf{Adv}^{\mathsf{ABE}}_{\Pi_{\mathsf{ABE}},\mathcal{B}}(\lambda). \tag{5}
$$

Finally, we show that it is computationally infeasible for $\mathcal{A}$ to win in $\mathbf{Game}_2$ if the ABE scheme Π_{ABE} satisfies IND-CPA security. For this purpose, we use $\mathcal{A}$ to construct a PPT adversary $\mathcal{D}$ that breaks IND-CPA security of Π_{ABE}. $\mathcal{D}$ interacts with $\mathcal{A}$ in the same way as $\mathcal{B}$ except the creation of the challenge ciphertext $\mathsf{ct}^*_{x^*}$. Upon $\mathcal{A}$'s challenge query on $(x^*, \mathsf{M}^*_0, \mathsf{M}^*_1)$, $\mathcal{D}$ makes the challenge query on $((x^*, 1, \mathsf{verk}^*), \mathsf{H}(\mathsf{M}^*_0), \mathsf{H}(\mathsf{M}^*_1))$ to $\mathsf{ABE}.\mathcal{C}$ and receives $\mathsf{ABE.ct}^*_{x^*,1,\mathsf{verk}^*}$. Here, $\mathsf{ABE.ct}^*_{x^*,1,\mathsf{verk}^*}$ are encryptions of $\mathsf{H}(\mathsf{M}^*_0)$ and $\mathsf{H}(\mathsf{M}^*_1)$ if $\mathsf{coin}' = 0$ and $\mathsf{coin}' = 1$, respectively. $\mathcal{D}$ chooses $\mathsf{M} \leftarrow_{\$} \mathcal{M}$ and runs $\mathsf{ABE.ct}^*_{x^*,0,\mathsf{verk}^*} \leftarrow \mathsf{ABE.Enc}(\mathsf{ABE.mpk}, (x^*, 0, \mathsf{verk}^*), \mathsf{M})$ and $\sigma^* \leftarrow \mathsf{Sig.Sign}(\mathsf{sigk}, [\mathsf{ABE.ct}^*_{x^*,0,\mathsf{verk}^*} \| \mathsf{ABE.ct}^*_{x^*,1,\mathsf{verk}^*}])$. $\mathcal{D}$ sets the challenge ciphertext $\mathsf{ct}^*_{x^*} = (\mathsf{verk}^*, \mathsf{ABE.ct}^*_{x^*,0,\mathsf{verk}^*}, \mathsf{ABE.ct}^*_{x^*,1,\mathsf{verk}^*}, \sigma^*)$, where $\mathsf{coin} = \mathsf{coin}'$. After $\mathcal{A}$ outputs $\widehat{\mathsf{coin}}$ as a guess of $\mathsf{coin} = \mathsf{coin}'$, $\mathcal{D}$ outputs $\widehat{\mathsf{coin}}' = \widehat{\mathsf{coin}}$ as a guess of coin' flipped by $\mathsf{ABE}.\mathcal{C}$.

$\mathcal{D}$ perfectly simulates $\mathbf{Game}_2$ by following essentially the same discussion as in $\mathcal{B}$ except the validity for answering trapdoor queries. In this proof, all $\mathcal{D}$'s Key extraction queries to answer $\mathcal{A}$'s trapdoor queries are valid since the definition of the Type-II adversaries ensures that $\mathsf{P}(x^*, y) = 0$ holds. We analyze the quantity of $|\Pr[W_2] - 1/2|$. Since $\mathsf{coin} = \mathsf{coin}'$ and $\widehat{\mathsf{coin}} = \widehat{\mathsf{coin}}'$, we have

$$
\begin{aligned}
\mathsf{Adv}^{\mathsf{ABE}}_{\Pi_{\mathsf{ABE}},\mathcal{D}}(\lambda) &= \left|\Pr\left[\widehat{\mathsf{coin}}' = \mathsf{coin}'\right] - \frac{1}{2}\right| \\
&= \left|\Pr\left[\widehat{\mathsf{coin}} = \mathsf{coin}\right] - \frac{1}{2}\right| \\
&= \left|\Pr[W_2] - \frac{1}{2}\right|.
\end{aligned}
$$

Therefore, we have

$$
\left|\Pr[W_2] - \frac{1}{2}\right| = \mathsf{Adv}^{\mathsf{ABE}}_{\Pi_{\mathsf{ABE}},\mathcal{D}}(\lambda). \tag{6}
$$

From (4) – (6), we have

$$
\begin{aligned}
\left|\Pr[W_0] - \frac{1}{2}\right| &\leq |\Pr[W_0] - \Pr[W_1]| + |\Pr[W_1] - \Pr[W_2]| + \left|\Pr[W_2] - \frac{1}{2}\right| \\
&\leq \mathsf{Adv}^{\mathsf{OTS}}_{\Gamma,\mathcal{F}}(\lambda) + 2\mathsf{Adv}^{\mathsf{ABE}}_{\Pi_{\mathsf{ABE}},\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathsf{ABE}}_{\Pi_{\mathsf{ABE}},\mathcal{D}}(\lambda).
\end{aligned}
$$

□

5 New Pair Encoding Scheme

In this section, we propose a delegatable transformation for a pair encoding scheme and a new pair encoding scheme for key-policy ABE for non-monotone span programs with compact ciphertexts. In Section 5.1, we review the definition of pair encoding. In Section 5.2, we propose a delegatable transformation. In Section 5.3, we propose a new pair encoding scheme.

5.1 Pair Encoding Scheme

In this section, we review a pair encoding scheme (PES) by following [Att14, AC16a, AC17b, Tak21].

Syntax. A PES for a predicate P consists of the following four polynomial time algorithms (Param, $\mathsf{EncC}, \mathsf{EncK}, \mathsf{Pair}$) defined as follows:

$\mathsf{Param}(\mathsf{par}) \to n$**:** On input par, Param outputs $n \in \mathbb{N}$ that specifies the number of common variables denoted by $\mathbf{b} := (b_1, \dots, b_n)$.

$\mathsf{EncC}(x, N) \to (w_1, w_2, \mathbf{c})$**:** On input $x \in \mathcal{X}$ and $N \in \mathbb{N}$, EncC outputs a vector of w_3 ciphertext-encoding polynomials $\mathbf{c} = (c_1, \dots, c_{w_3})$ in non-lone ciphertext-encoding variables s_0 and $\mathbf{s} = (s_1, \dots, s_{w_1})$ and lone ciphertext-encoding variables $\hat{\mathbf{s}} = (\hat{s}_1, \dots, \hat{s}_{w_2})$. The ℓ-th polynomial is given by

$$c_\ell := \sum_{z \in [w_2]} \eta_{\ell,z} \hat{s}_z + \sum_{i \in [0,w_1], j \in [n]} \eta_{\ell,i,j} s_i b_j$$

for $\ell \in [w_3]$, where $\eta_{\ell,z}, \eta_{\ell,i,j} \in \mathbb{Z}_N$.

$\mathsf{EncK}(y, N) \to (m_1, m_2, \mathbf{k})$**:** On input $y \in \mathcal{Y}$ and $N \in \mathbb{N}$, EncK outputs a vector of m_3 key-encoding polynomials $\mathbf{k} = (k_1, \dots, k_{m_3})$ in non-lone key-encoding variables $\mathbf{r} = (r_1, \dots, r_{m_1})$ and lone key-encoding variables α and $\hat{\mathbf{r}} = (\hat{r}_1, \dots, \hat{r}_{m_2})$. The t-th polynomial is given by

$$k_t := \phi_t \alpha + \sum_{z' \in [m_2]} \phi_{t,z'} \hat{r}_{z'} + \sum_{i' \in [m_1], j \in [n]} \phi_{t,i',j} r_{i'} b_j$$

for $t \in [m_3]$, where $\phi_t, \phi_{t,z'}, \phi_{t,i',j} \in \mathbb{Z}_N$.

$\mathsf{Pair}(x, y, N) \to (\mathbf{E}, \overline{\mathbf{E}})$**:** On input $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and $N \in \mathbb{N}$, Pair outputs two matrices $\mathbf{E}$ and $\overline{\mathbf{E}}$ of size $(w_1 + 1) \times m_3$ and $w_3 \times m_1$, respectively.

Remark 4. A predicate encoding is a special case of pair encoding, where both the numbers of non-lone ciphertext-encoding variable w_1 and key-encoding variables m_1 are always one.

Correctness. A PES for a predicate P is correct if for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ such that $\mathsf{P}(x, y) = 1$, it holds that

$$\mathbf{s}^\top \mathbf{E} \mathbf{k} + \mathbf{c}^\top \overline{\mathbf{E}} \mathbf{r} = \sum_{i \in [0,w_1], t \in [m_3]} s_i E_{i,t} k_t + \sum_{\ell \in [w_3], i' \in [m_1]} c_\ell \overline{E}_{\ell,i'} r_{i'} = \alpha s_0. \tag{7}$$

Security. We review the definitions of perfect security, relaxed perfect security, and symbolic security. For this purpose, we may use the notation $\mathbf{c}(s_0, \mathbf{s}, \hat{\mathbf{s}}, \mathbf{b})$ and $\mathbf{k}(\alpha, \mathbf{r}, \hat{\mathbf{r}}, \mathbf{b})$ to specify variables for creating key-encoding polynomials $\mathbf{k}$ and ciphertext-encoding polynomials $\mathbf{c}$, respectively. Furthermore, for $d \in [m_2]$, let $\mathbf{k}_d(r_d, \mathbf{b})$ denote $\mathbf{k}(\alpha, \mathbf{r}, \hat{\mathbf{r}}, \mathbf{b})$ except that $\alpha = 0$, $r_d = 0$ for $d \in [m_1] \setminus \{d\}$, and $\hat{\mathbf{r}} = \mathbf{0}$. Moreover, we use the following randomized polynomial time algorithm Samp to review the definition of relaxed perfect security.

$\mathsf{Samp}(d, x, y, N) \to \mathbf{b}_d := (b_{d,1}, \dots, b_{d,n})$**:** This algorithm takes an index for non-lone key-encoding variables $d \in [m_1]$, $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and $N \in \mathbb{N}$ as input, and outputs a sequence of n numbers in $\mathbb{Z}_N$. We require that the probability of this algorithm outputs $(u \cdot b_{d,1}, \dots, u \cdot b_{d,n})$ is equal to the probability that it outputs $(b_{d,1}, \dots, b_{d,n})$ for any $u \in \mathbb{Z}_N^*$.

Definition 5.1 (Perfect Security [Att14])**.** A $\mathsf{PES} = (\mathsf{Param}, \mathsf{EncK}, \mathsf{EncC}, \mathsf{Pair})$ for a predicate P satisfies perfect security if for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ such that $\mathsf{P}(x, y) = 0$, it holds that

$$(s_0, \mathbf{s}, \mathbf{r}, \mathbf{c}(s_0, \mathbf{s}, \hat{\mathbf{s}}, \mathbf{b}), \mathbf{k}(0, \mathbf{r}, \hat{\mathbf{r}}, \mathbf{b})) \equiv (s_0, \mathbf{s}, \mathbf{r}, \mathbf{c}(s_0, \mathbf{s}, \hat{\mathbf{s}}, \mathbf{b}), \mathbf{k}(\boxed{\alpha}, \mathbf{r}, \hat{\mathbf{r}}, \mathbf{b})) \tag{8}$$

where $s_0 \leftarrow_{\$} \mathbb{Z}_N$, $\mathbf{s} \leftarrow_{\$} \mathbb{Z}_N^{w_1}$, $\mathbf{r} \leftarrow_{\$} \mathbb{Z}_N^{m_1}$, $\hat{\mathbf{s}} \leftarrow_{\$} \mathbb{Z}_N^{w_2}$, $\hat{\mathbf{r}} \leftarrow_{\$} \mathbb{Z}_N^{m_2}$, $\mathbf{b} \leftarrow_{\$} \mathbb{Z}_N^n$, and $\alpha \leftarrow_{\$} \mathbb{Z}_N$.

Theorem 5.1 ([Att14, AC16b, Tak21])**.** If there is a $\mathsf{PES} = (\mathsf{Param}, \mathsf{EncK}, \mathsf{EncC}, \mathsf{Pair})$ for a predicate P satisfying the perfect security, there is an adaptively secure ABE scheme for the same predicate P under the standard k-linear assumption.

Remark 5. Our generic construction of ABEET requires three-level delegatable ABE whose second and third levels support only the equality predicate as in identity-based encryption. A pair encoding scheme for identity-based encryption satisfies perfect security.

Definition 5.2 (Relaxed Perfect Security [AC16b])**.** For a $\mathsf{PES} = (\mathsf{Param}, \mathsf{EncK}, \mathsf{EncC}, \mathsf{Pair})$ for a predicate P satisfies relaxed perfect security if there exists a PPT algorithm Samp such that for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ such that $\mathsf{P}(x, y) = 0$, and all $d \in [m_1]$, it holds that

$$\{s_0, \mathbf{s}, r_d, \mathbf{c}(s_0, \mathbf{s}, \hat{\mathbf{s}}, \mathbf{b}), \mathbf{k}_d(r_d, \mathbf{b})\} \approx \left\{s_0, \mathbf{s}, r_d, \mathbf{c}(s_0, \mathbf{s}, \hat{\mathbf{s}}, \mathbf{b}), \mathbf{k}_d(r_d, \mathbf{b} + \boxed{\mathbf{b}_d})\right\} \tag{9}$$

where $s_0 \leftarrow_{\$} \mathbb{Z}_N$, $\mathbf{s} \leftarrow_{\$} \mathbb{Z}_N^{w_1}$, $r_d \leftarrow_{\$} \mathbb{Z}_N$, $\hat{\mathbf{s}} \leftarrow_{\$} \mathbb{Z}_N^{w_2}$, $\mathbf{b} \leftarrow_{\$} \mathbb{Z}_N^n$, and $\mathbf{b}_d \leftarrow \mathsf{Samp}(d, x, y, N)$. Furthermore, it holds that

$$\begin{aligned} &\left\{s_0, \mathbf{s}, \mathbf{r}, \mathbf{c}(s_0, \mathbf{s}, \hat{\mathbf{s}}, \mathbf{b}), \mathbf{k}(0, \mathbf{0}, \hat{\mathbf{r}}, \mathbf{0}) + \sum_{d \in [m_1]} \mathbf{k}_d(r_d, \mathbf{b} + \mathbf{b}_d)\right\} \\ \approx &\left\{s_0, \mathbf{s}, \mathbf{r}, \mathbf{c}(s_0, \mathbf{s}, \hat{\mathbf{s}}, \mathbf{b}), \mathbf{k}(\boxed{\alpha}, \mathbf{0}, \hat{\mathbf{r}}, \mathbf{0}) + \sum_{d \in [m_1]} \mathbf{k}_d(r_d, \mathbf{b} + \mathbf{b}_d)\right\}, \end{aligned} \tag{10}$$

where $s_0 \leftarrow_{\$} \mathbb{Z}_N$, $\mathbf{s} \leftarrow_{\$} \mathbb{Z}_N^{w_1}$, $\mathbf{r} \leftarrow_{\$} \mathbb{Z}_N^{m_1}$, $\hat{\mathbf{s}} \leftarrow_{\$} \mathbb{Z}_N^{w_2}$, $\hat{\mathbf{r}} \leftarrow_{\$} \mathbb{Z}_N^{m_2}$, $\mathbf{b} \leftarrow_{\$} \mathbb{Z}_N^n$, $\alpha \leftarrow_{\$} \mathbb{Z}_N$, $\mathbf{b}_d \leftarrow \mathsf{Samp}(d, x, y, N)$ for $d \in [m_1]$.

Theorem 5.2 ([AC16b, Tak21])**.** If there is a $\mathsf{PES} = (\mathsf{Param}, \mathsf{EncK}, \mathsf{EncC}, \mathsf{Pair})$ for a predicate P satisfying the relaxed perfect security, there is a semi-adaptively secure ABE scheme for the same predicate P under the standard k-linear assumption.

Remark 6. If PES satisfies the perfect security, it also satisfies the relaxed perfect security by setting outputs of Samp as zero vectors.

Definition 5.3 (Symbolic Security [AC17b])**.** A $\mathsf{PES} = (\mathsf{Param}, \mathsf{EncK}, \mathsf{EncC}, \mathsf{Pair})$ for a predicate P satisfies (d_1, d_2)-selective symbolic security for positive integers d_1 and d_2 if there exist three deterministic polynomial-time algorithms EncB, EncS, and EncR such that for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ such that $\mathsf{P}(x, y) = 0$,

- $\mathsf{EncB}(x) \to (\mathbf{B}_1, \ldots, \mathbf{B}_n) \in (\mathbb{Z}_N^{d_1 \times d_2})^n$;
- $\mathsf{EncR}(x, y) \to (\mathbf{r}_1, \ldots, \mathbf{r}_{m_1}, \mathbf{a}, \hat{\mathbf{r}}_1, \ldots, \hat{\mathbf{r}}_{m_2}) \in (\mathbb{Z}_N^{d_1})^{m_1} \times (\mathbb{Z}_N^{d_2})^{m_2+1}$;
- $\mathsf{EncS}(x) \to (\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{w_1}, \hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_{w_2}) \in (\mathbb{Z}_N^{d_2})^{w_1+1} \times (\mathbb{Z}_N^{d_1})^{w_2}$;

such that $\langle \mathbf{s}_0, \mathbf{a} \rangle \neq 0$, and if we substitute

$$s_i : \mathbf{s}_i^\top, \qquad \hat{s}_i : \hat{\mathbf{s}}_i^\top, \qquad s_i b_j : \mathbf{B}_j \mathbf{s}_i^\top, \qquad r_{i'} : \mathbf{r}_{i'}, \qquad \alpha : \mathbf{a}, \qquad \hat{r}_{i'} : \hat{\mathbf{r}}_{i'}, \qquad r_{i'} b_j : \mathbf{r}_{i'} \mathbf{B}_j,$$

for $z \in [w_2], i \in [0, w_1], j \in [n], z' \in [m_2]$, and $i' \in [m_1]$ in all key-encoding polynomials output by $\mathsf{EncK}(y, N)$ and all ciphertext-encoding polynomials output by $\mathsf{EncC}(x, N)$, then they evaluate to $\mathbf{0}$.

Similarly, the PES satisfies (d_1, d_2)-co-selective symbolic security if there exist EncB, EncR, and EncS as above except that inputs of these three algorithms are y, (x, y), and y, respectively. Finally, the PES satisfies (d_1, d_2)-symbolic security if it satisfies (d_1', d_2')-selective symbolic security such that $d_1' \leq d_1, d_2' \leq d_2$ and (d_1'', d_2'')-selective symbolic security such that $d_1'' \leq d_1, d_2'' \leq d_2$.

Theorem 5.3 ([AC17b])**.** If there is a $\mathsf{PES} = (\mathsf{Param}, \mathsf{EncC}, \mathsf{EncK}, \mathsf{Pair})$ for a predicate P satisfying the symbolic security, there is an adaptively secure ABE scheme for the same predicate P under the q-ratio assumption.

Remark 7. As Agrawal and Chase [AC17b] claimed, if PES satisfies the perfect security or the relaxed perfect security, it also satisfies the symbolic security with a mild modification.

5.2 Delegatable Transformation

We show how to combine several pair encoding schemes to be delegatable one. Specifically, let $\mathsf{PES}^{(\ell)} = (\mathsf{Param}^{(\ell)}, \mathsf{EncC}^{(\ell)}, \mathsf{EncK}^{(\ell)}, \mathsf{Pair}^{(\ell)})$ for $\ell \in [L]$ denote pair encoding schemes for predicates $\mathsf{P}^{(\ell)} : \mathcal{X}^{(\ell)} \times \mathcal{Y}^{(\ell)} \to \{0, 1\}$, respectively. Hereafter, any values with superscripts (ℓ) denote those for $\mathsf{PES}^{(\ell)}$, e.g., $n^{(\ell)}, w_1^{(\ell)}, m_1^{(\ell)}$, and so on. We set $\mathcal{X} = \mathcal{X}^{(1)} \times \cdots \times \mathcal{X}^{(L)}$ and $\mathcal{Y} = \mathcal{Y}^{(1)} \times \cdots \times \mathcal{Y}^{(L)}$. Based on them, the goal is constructing $\mathsf{PES} = (\mathsf{Param}, \mathsf{EncK}, \mathsf{EncC}, \mathsf{Pair})$ for a delegatable predicate $\mathsf{P} : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ defined as follows.

- For $x = (x^{(1)}, \ldots, x^{(L)}) \in \mathcal{X}$ (resp. $y = (y^{(1)}, \ldots, y^{(L)}) \in \mathcal{Y}$), some $x^{(\ell)}$ (resp. $y^{(\ell)}$) may not be elements of $\mathcal{X}^{(\ell)}$ (resp. $\mathcal{Y}^{(\ell)}$) but empty denoted by $\perp$. Let $L_x \subseteq [L]$ (resp. $L_y \subseteq [L]$) denote a set of indices such that $x^{(\ell)} \neq \perp$ hold for all $\ell \in L_x$ (resp. $y^{(\ell)} \neq \perp$ hold for all $\ell \in L_y$). Moreover, we define $\ell_y \in L_y$ such that $\ell_y \leq \ell$ holds for all $\ell \in L_y$.
- For $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, it holds that $\mathsf{P}(x, y) = 1$ iff $L_y \subseteq L_x$ holds and $\mathsf{P}^{(\ell)}(x^{(\ell)}, y^{(\ell)}) = 1$ hold for all $\ell \in L_y$.

To handle L predicates $\mathsf{P}^{(1)}, \ldots, \mathsf{P}^{(L)}$ simultaneously for PES, we use $n = \sum_{\ell=1}^{L} n^{(\ell)}$ common variables $(\mathbf{b}^{(1)}, \ldots, \mathbf{b}^{(L)})$. Briefly speaking, ciphertext-encoding polynomials $\mathbf{c}$ for $x \in \mathcal{X}$ are concatenations of $\mathbf{c}^{(\ell)}$ for $(x_\ell)_{\ell \in L_x}$, while key-encoding polynomials $\mathbf{k}$ for $y \in \mathcal{Y}$ are concatenations of $\mathbf{k}^{(\ell)}$ for $(y_\ell)_{\ell \in L_y}$ with auxiliary polynomials depending on $(\mathbf{b}^{(\ell)})_{\ell \in [L] \setminus L_y}$ to realize key delegation. To satisfy both correctness and security, the polynomials satisfy the following condition:

- Ciphertext-encoding polynomials and key-encoding polynomials for a predicate $\mathsf{P}^{(\ell)}$ depends only on $\mathbf{b}^{(\ell)}$ as $\mathsf{PES}^{(\ell)}$.
- All $\mathbf{c}^{(\ell)}$ in the same $\mathbf{c}$ share the same non-lone ciphertext-encoding variables s_0 and $\mathbf{s}$. Similarly, all $\mathbf{k}^{(\ell)}$ in the same $\mathbf{k}$ share the same non-lone key-encoding variables $\mathbf{r}$.
- All $\mathbf{c}^{(\ell)}$ in the same $\mathbf{c}$ use distinct lone ciphertext-encoding variables $\hat{\mathbf{s}}^{(\ell)}$. Similarly, all $\mathbf{k}^{(\ell)}$ in the same $\mathbf{k}$ use distinct lone key-encoding variables $\alpha^{(\ell)}$ and $\hat{\mathbf{r}}^{(\ell)}$, where it holds that $\alpha = \sum_{\ell \in L_y} \alpha^{(\ell)}$.

We define $\mathsf{PES} = (\mathsf{Param}, \mathsf{EncK}, \mathsf{EncC}, \mathsf{Pair})$ for a delegatable predicate P as follows.

$\mathsf{Param}(\mathsf{par})$**:** On input $\mathsf{par} = (\mathsf{par}^{(\ell)})_{\ell\in[L]}$, Param runs $n^{(\ell)} \leftarrow \mathsf{Param}^{(\ell)}(\mathsf{par}^{(\ell)})$ for $\ell \in [L]$ and outputs $n := \sum_{\ell\in[L]} n^{(\ell)}$ that specifies the number of common variables denoted by $\mathbf{b} := (\mathbf{b}^{(\ell)})_{\ell\in[L]}$.

$\mathsf{EncC}(x, N)$**:** On input $x \in \mathcal{X}$ and $N \in \mathbb{N}$, set $w_1 := \max_{\ell\in L_x} w_1^{(\ell)}$ and $w_2 := \sum_{\ell\in L_x} w_2^{(\ell)}$. EncC runs $(w_1^{(\ell)}, w_2^{(\ell)}, \mathbf{c}^{(\ell)}(s_0, (s_1, \dots, s_{w_1^{(\ell)}}), \hat{\mathbf{s}}^{(\ell)}, \mathbf{b}^{(\ell)})) \leftarrow \mathsf{EncC}(x^{(\ell)}, N)$ for $\ell \in L_x$ and outputs a vector of $w_3 := \sum_{\ell\in L_x} w_3^{(\ell)}$ ciphertext-encoding polynomials

$$\mathbf{c}(s_0, \mathbf{s}, \hat{\mathbf{s}}, \mathbf{b}) := (\mathbf{c}^{(\ell)}(s_0, (s_1, \dots, s_{w_1^{(\ell)}}), \hat{\mathbf{s}}^{(\ell)}, \mathbf{b}^{(\ell)}))_{\ell\in L_x}$$

in non-lone ciphertext-encoding variables s_0 and $\mathbf{s}$ and lone ciphertext-encoding variables $\hat{\mathbf{s}} := (\hat{\mathbf{s}}^{(\ell)})_{\ell\in L_x}$.

$\mathsf{EncK}(y, N) \to (m_1, m_2, \mathbf{k})$**:** On input $y \in \mathcal{Y}$ and $N \in \mathbb{N}$, set $m_1 := \max_{\ell\in L_y} m_1^{(\ell)}$ and $m_2 := \sum_{\ell\in L_y} m_2^{(\ell)}$. EncK runs $(m_1^{(\ell)}, m_2^{(\ell)}, \mathbf{k}^{(\ell)}(\alpha^{(\ell)}, (r_1, \dots, r_{m_1^{(\ell)}}), \hat{\mathbf{r}}^{(\ell)}, \mathbf{b}^{(\ell)})) \leftarrow \mathsf{EncK}^{(\ell)}(y^{(\ell)}, N)$ for $\ell \in L_y$ such that $\alpha^{(\ell_y)} = \alpha - \sum_{\ell\in L_y\setminus\{\ell_y\}} \alpha^{(\ell)}$ and outputs a vector of $m_3 := \sum_{\ell\in L_y} m_3^{(\ell)} + m_1' \sum_{\ell\in[L]\setminus L_y} n^{(\ell)}$ key-encoding polynomials

$$\mathbf{k}(\alpha, \mathbf{r}, \hat{\mathbf{r}}, \mathbf{b}) := \left((\mathbf{k}^{(\ell)}(\alpha^{(\ell)}, (r_1, \dots, r_{m_1^{(\ell)}}), \hat{\mathbf{r}}^{(\ell)}, \mathbf{b}^{(\ell)}))_{\ell\in L_y}, ((r_{i'} b_j^{(\ell)})_{i'\in[m_1'], j\in[n^{(\ell)}]})_{\ell\in[L]\setminus L_y}\right),$$

where $m_1' := \min\{m_1, \max_{\ell\in[L]\setminus L_y} m_1^{(\ell)}\}$, in non-lone key-encoding variables $\mathbf{r}$ and lone key-encoding variables $\hat{\mathbf{r}} := ((\alpha^{(\ell)})_{\ell\in L_y\setminus\{y_\ell\}}, (\hat{\mathbf{r}}^{(\ell)})_{\ell\in L_y})$.

Correctness. We did not describe Pair since it may be complicated. In turn, we describe how to recover αs_0 if $\mathsf{P}(x, y) = 1$ holds since it should be simpler to understand. Due to the correctness of $\mathsf{PES}^{(\ell)}$ for $\ell \in L_y$, since $L_y \subseteq L_x$ holds, we can recover $\alpha^{(\ell)} s_0$ for $\ell \in [L_y]$ from $(\mathbf{c}^{(\ell)}(s_0, (s_1, \dots, s_{w_1^{(\ell)}}), \hat{\mathbf{s}}, \mathbf{b}^{(\ell)}), \mathbf{k}^{(\ell)}(\alpha^{(\ell)}, (r_1, \dots, r_{m_1^{(\ell)}}), \hat{\mathbf{r}}^{(\ell)}, \mathbf{b}^{(\ell)}))$ for $\ell \in L_y$, respectively. Then, we can recover $\sum_{\ell\in L_y} \alpha^{(\ell)} s_0 = \alpha s_0$.

Remark 8. Although we omit the detailed description, the above PES is obviously delegatable. In particular, given $\mathbf{k}$ for $y \in \mathcal{Y}$, we can compute $\mathbf{k}'$ for $y' \in \mathcal{Y}$ without changing α if $y^{(\ell)} = y'^{(\ell)}$ holds for all $\ell \in L_y$. A point to note is that, to share the same non-lone key-encoding variables between $\ell \in L_y$ and $\ell \in L_{y'} \setminus L_y$, $\mathbf{k}$ contains $((r_{i'} b_j^{(\ell)})_{i'\in[m_1'], j\in[n^{(\ell)}]})_{\ell\in[L]\setminus L_y}$.

Security. We show that the PES preserves the security of $\mathsf{PES}^{(\ell)}$ for $\ell \in [L]$ as stated in Theorems 5.4–5.6.

Theorem 5.4. A $\mathsf{PES} = (\mathsf{Param}, \mathsf{EncC}, \mathsf{EncK}, \mathsf{Pair})$ for a predicate P described above satisfies the perfect security when $\mathsf{PES}^{(\ell)}$ for $\ell \in [L]$ also satisfy the perfect security.

Proof. If $L_x \subset L_y$, it is obvious that PES satisfies the perfect security. Otherwise, i.e., $L_y \subseteq L_x$, we show that PES satisfies the perfect security (8) if there is an index $\ell^\star$ such that $\mathsf{P}^{(\ell^\star)}(x^{(\ell^\star)}, y^{(\ell^\star)}) = 0$. For simplicity, we consider the case that $\ell^\star \neq \ell_y$. The proof for the other case is essentially the same.

We first observe that the left and right hand sides of (8) satisfy that

$$
\begin{aligned}
&\{s_0, \mathbf{s}, \mathbf{r}, \mathbf{c}(s_0, \mathbf{s}, \hat{\mathbf{s}}, \mathbf{b}), \mathbf{k}(0, \mathbf{r}, \hat{\mathbf{r}}, \mathbf{b})\} \\
&= \left\{ \begin{array}{c} s_0, \mathbf{s}, \mathbf{r}, (\mathbf{c}^{(\ell)}(s_0, (s_1, \dots, s_{w_1^{(\ell)}}), \hat{\mathbf{s}}^{(\ell)}, \mathbf{b}^{(\ell)}))_{\ell \in L_x}, \\ \mathbf{k}^{(\ell_y)}(-\sum_{\ell \in L_y \setminus \{\ell_y\}} \alpha^{(\ell)}, (r_1, \dots, r_{m_1^{(\ell_y)}}), \hat{\mathbf{r}}^{(\ell_y)}, \mathbf{b}^{(\ell_y)}), \\ (\mathbf{k}^{(\ell)}(\alpha^{(\ell)}, (r_1, \dots, r_{m_1^{(\ell)}}), \hat{\mathbf{r}}^{(\ell)}, \mathbf{b}^{(\ell)}))_{\ell \in L_y \setminus \{\ell_y\}}, ((r_{i'} b_j^{(\ell)})_{i' \in [m_1'], j \in [n^{(\ell)}]})_{\ell \in [L] \setminus L_y} \end{array} \right\} \\
&\equiv \left\{ \begin{array}{c} s_0, \mathbf{s}, \mathbf{r}, (\mathbf{c}^{(\ell)}(s_0, (s_1, \dots, s_{w_1^{(\ell)}}), \hat{\mathbf{s}}^{(\ell)}, \mathbf{b}^{(\ell)}))_{\ell \in L_x}, \\ \mathbf{k}^{(\ell^\star)}(-\sum_{\ell \in L_y \setminus \{\ell^\star\}} \alpha^{(\ell)}, (r_1, \dots, r_{m_1^{(\ell^\star)}}), \hat{\mathbf{r}}^{(\ell^\star)}, \mathbf{b}^{(\ell^\star)}), \\ (\mathbf{k}^{(\ell)}(\alpha^{(\ell)}, (r_1, \dots, r_{m_1^{(\ell)}}), \hat{\mathbf{r}}^{(\ell)}, \mathbf{b}^{(\ell)}))_{\ell \in L_y \setminus \{\ell^\star\}}, ((r_{i'} b_j^{(\ell)})_{i' \in [m_1'], j \in [n^{(\ell)}]})_{\ell \in [L] \setminus L_y} \end{array} \right\} \\
&= \left\{ \begin{array}{c} s_0, \mathbf{s}, \mathbf{r}, (\mathbf{c}^{(\ell)}(s_0, (s_1, \dots, s_{w_1^{(\ell)}}), \hat{\mathbf{s}}^{(\ell)}, \mathbf{b}^{(\ell)}))_{\ell \in L_x}, \\ \mathbf{k}^{(\ell^\star)}(0, (r_1, \dots, r_{m_1^{(\ell^\star)}}), \hat{\mathbf{r}}^{(\ell^\star)}, \mathbf{b}^{(\ell^\star)}) + \mathbf{k}^{(\ell^\star)}(-\sum_{\ell \in L_y \setminus \{\ell^\star\}} \alpha^{(\ell)}, \mathbf{0}, \mathbf{0}, \mathbf{0}), \\ (\mathbf{k}^{(\ell)}(\alpha^{(\ell)}, (r_1, \dots, r_{m_1^{(\ell)}}), \hat{\mathbf{r}}^{(\ell)}, \mathbf{b}^{(\ell)}))_{\ell \in L_y \setminus \{\ell^\star\}}, ((r_{i'} b_j^{(\ell)})_{i' \in [m_1'], j \in [n^{(\ell)}]})_{\ell \in [L] \setminus L_y} \end{array} \right\}
\end{aligned}
$$

and

$$
\begin{aligned}
&\{s_0, \mathbf{s}, \mathbf{r}, \mathbf{c}(s_0, \mathbf{s}, \hat{\mathbf{s}}, \mathbf{b}), \mathbf{k}(\alpha, \mathbf{r}, \hat{\mathbf{r}}, \mathbf{b})\} \\
&= \left\{ \begin{array}{c} s_0, \mathbf{s}, \mathbf{r}, (\mathbf{c}^{(\ell)}(s_0, (s_1, \dots, s_{w_1^{(\ell)}}), \hat{\mathbf{s}}^{(\ell)}, \mathbf{b}^{(\ell)}))_{\ell \in L_x}, \\ \mathbf{k}^{(\ell_y)}(\alpha - \sum_{\ell \in L_y \setminus \{\ell_y\}} \alpha^{(\ell)}, (r_1, \dots, r_{m_1^{(\ell_y)}}), \hat{\mathbf{r}}^{(\ell_y)}, \mathbf{b}^{(\ell_y)}), \\ (\mathbf{k}^{(\ell)}(\alpha^{(\ell)}, (r_1, \dots, r_{m_1^{(\ell)}}), \hat{\mathbf{r}}^{(\ell)}, \mathbf{b}^{(\ell)}))_{\ell \in L_y \setminus \{\ell_y\}}, ((r_{i'} b_j^{(\ell)})_{i' \in [m_1'], j \in [n^{(\ell)}]})_{\ell \in [L] \setminus L_y} \end{array} \right\} \\
&\equiv \left\{ \begin{array}{c} s_0, \mathbf{s}, \mathbf{r}, (\mathbf{c}^{(\ell)}(s_0, (s_1, \dots, s_{w_1^{(\ell)}}), \hat{\mathbf{s}}^{(\ell)}, \mathbf{b}^{(\ell)}))_{\ell \in L_x}, \\ \mathbf{k}^{(\ell^\star)}(\alpha - \sum_{\ell \in L_y \setminus \{\ell^\star\}} \alpha^{(\ell)}, (r_1, \dots, r_{m_1^{(\ell^\star)}}), \hat{\mathbf{r}}^{(\ell^\star)}, \mathbf{b}^{(\ell^\star)}), \\ (\mathbf{k}^{(\ell)}(\alpha^{(\ell)}, (r_1, \dots, r_{m_1^{(\ell)}}), \hat{\mathbf{r}}^{(\ell)}, \mathbf{b}^{(\ell)}))_{\ell \in L_y \setminus \{\ell^\star\}}, ((r_{i'} b_j^{(\ell)})_{i' \in [m_1'], j \in [n^{(\ell)}]})_{\ell \in [L] \setminus L_y} \end{array} \right\} \\
&= \left\{ \begin{array}{c} s_0, \mathbf{s}, \mathbf{r}, (\mathbf{c}^{(\ell)}(s_0, (s_1, \dots, s_{w_1^{(\ell)}}), \hat{\mathbf{s}}^{(\ell)}, \mathbf{b}^{(\ell)}))_{\ell \in L_x}, \\ \mathbf{k}^{(\ell^\star)}(\alpha, (r_1, \dots, r_{m_1^{(\ell^\star)}}), \hat{\mathbf{r}}^{(\ell^\star)}, \mathbf{b}^{(\ell^\star)}) + \mathbf{k}^{(\ell^\star)}(\alpha - \sum_{\ell \in L_y \setminus \{\ell^\star\}} \alpha^{(\ell)}, \mathbf{0}, \mathbf{0}, \mathbf{0}), \\ (\mathbf{k}^{(\ell)}(\alpha^{(\ell)}, (r_1, \dots, r_{m_1^{(\ell)}}), \hat{\mathbf{r}}^{(\ell)}, \mathbf{b}^{(\ell)}))_{\ell \in L_y \setminus \{\ell^\star\}}, ((r_{i'} b_j^{(\ell)})_{i' \in [m_1'], j \in [n^{(\ell)}]})_{\ell \in [L] \setminus L_y} \end{array} \right\}
\end{aligned}
$$

where $s_0 \leftarrow_{\$} \mathbb{Z}_N$, $\mathbf{s} \leftarrow_{\$} \mathbb{Z}_N^{w_1}$, $(\mathbf{r}^{(\ell)})_{\ell \in L_y} \leftarrow_{\$} \mathbb{Z}_N^{m_1}$, $(\hat{\mathbf{s}}^{(\ell)})_{\ell \in L_x} \leftarrow_{\$} \mathbb{Z}_N^{w_2}$, $(\hat{\mathbf{r}}^{(\ell)})_{\ell \in L_y} \leftarrow_{\$} \mathbb{Z}_N^{m_2}$, $\mathbf{b} \leftarrow_{\$} \mathbb{Z}_N^n$, $(\alpha^{(\ell)})_{\ell \in L_y} \leftarrow_{\$} \mathbb{Z}_N^{|L_y|}$, and $\alpha \leftarrow_{\$} \mathbb{Z}_N$. Thus, to prove the perfect security (8) of PES, it is sufficient to show that

$$
\begin{aligned}
&\{s_0, \mathbf{s}, \mathbf{r}, \mathbf{c}^{(\ell^\star)}(s_0, (s_1, \dots, s_{w_1^{(\ell^\star)}}), \hat{\mathbf{s}}^{(\ell^\star)}, \mathbf{b}^{(\ell^\star)}), \mathbf{k}^{(\ell^\star)}(0, (r_1, \dots, r_{m_1^{(\ell^\star)}}), \hat{\mathbf{r}}^{(\ell^\star)}, \mathbf{b}^{(\ell^\star)})\} \\
&\equiv \{s_0, \mathbf{s}, \mathbf{r}, \mathbf{c}^{(\ell^\star)}(s_0, (s_1, \dots, s_{w_1^{(\ell^\star)}}), \hat{\mathbf{s}}^{(\ell^\star)}, \mathbf{b}^{(\ell^\star)}), \mathbf{k}^{(\ell^\star)}(\alpha, (r_1, \dots, r_{m_1^{(\ell^\star)}}), \hat{\mathbf{r}}^{(\ell^\star)}, \mathbf{b}^{(\ell^\star)})\}
\end{aligned}
$$

holds, where $s_0 \leftarrow_{\$} \mathbb{Z}_N$, $(s_1, \dots, s_{w_1^{(\ell^\star)}}) \leftarrow_{\$} \mathbb{Z}_N^{w_1^{(\ell^\star)}}$, $(r_1, \dots, r_{m_1^{(\ell^\star)}}) \leftarrow_{\$} \mathbb{Z}_N^{m_1^{(\ell^\star)}}$, $\hat{\mathbf{s}}^{(\ell^\star)} \leftarrow_{\$} \mathbb{Z}_N^{w_2^{(\ell^\star)}}$, $\hat{\mathbf{r}}^{(\ell^\star)} \leftarrow_{\$} \mathbb{Z}_N^{m_2^{(\ell^\star)}}$, $\mathbf{b}^{(\ell^\star)} \leftarrow_{\$} \mathbb{Z}_N^{n^{(\ell^\star)}}$, and $\alpha \leftarrow_{\$} \mathbb{Z}_N$. Since the statistical equivalence is exactly the perfect security of $\mathsf{PES}^{(\ell^\star)}$, the perfect security of PES holds. □

Theorem 5.5. A $\mathsf{PES} = (\mathsf{Param}, \mathsf{EncC}, \mathsf{EncK}, \mathsf{Pair})$ for a predicate P described above satisfies the relaxed perfect security when $\mathsf{PES}^{(\ell)}$ for $\ell \in [L]$ also satisfy the relaxed perfect security.

Proof. If $L_x \subset L_y$, PES satisfies the perfect security. Thus, PES also satisfies the relaxed perfect security from Remark 6. Otherwise, i.e., $L_y \subseteq L_x$, we show that PES satisfies the relaxed perfect security (9) and (10) if there is an index $\ell^\star$ such that $\mathsf{P}^{(\ell^\star)}(x^{(\ell^\star)}, y^{(\ell^\star)}) = 0$. For simplicity, we consider the case that $\ell^\star \neq \ell_y$. The proof for the other case is essentially the same.

At first, we prove (9). We set outputs of $\mathsf{Samp}(d, x, y, N)$ as $\mathbf{0}$ if $d > m_1^{(\ell^\star)}$ holds, i.e., key-encoding polynomials $\mathbf{k}^{(\ell^\star)}(\alpha^{(\ell^\star)}, (r_1, \dots, r_{m_1^{(\ell^\star)}}), \hat{\mathbf{r}}^{(\ell^\star)}, \mathbf{b}^{(\ell^\star)})$ does not depend on a non-lone key-encoding variable r_d. In this case, left and right hand sides of (9) are same. In contrast, if $d \leq m_1^{(\ell^\star)}$ holds, i.e., key-encoding polynomials $\mathbf{k}^{(\ell^\star)}(\alpha^{(\ell^\star)}, (r_1, \dots, r_{m_1^{(\ell^\star)}}), \hat{\mathbf{r}}^{(\ell^\star)}, \mathbf{b}^{(\ell^\star)})$ depend on a non-lone key-encoding variable r_d, we set outputs $(\mathbf{b}_d^{(\ell)})_{\ell \in [L]} \leftarrow \mathsf{Samp}(d, x, y, N)$ so that $\mathbf{b}_d^{(\ell)} = \mathbf{0}$ if $\ell \in [L] \setminus \{\ell^\star\}$ and $\mathbf{b}_d^{(\ell^\star)} \leftarrow \mathsf{Samp}(d, x^{(\ell^\star)}, y^{(\ell^\star)}, N)$ otherwise. In this case, the relaxed perfect security (9) of $\mathsf{PES}^{(\ell^\star)}$ ensures that of PES.

Next, we prove (10). We first observe that the left and right hand sides of (10) satisfy that

$$
\begin{aligned}
&\left\{ s_0, \mathbf{s}, \mathbf{r}, \mathbf{c}(s_0, \mathbf{s}, \hat{\mathbf{s}}, \mathbf{b}), \mathbf{k}(0, \mathbf{0}, \hat{\mathbf{r}}, 0) + \sum_{d \in [m_1]} \mathbf{k}_d(r_d, \mathbf{b} + \mathbf{b}_d) \right\} \\
&= \left\{ \begin{array}{c} s_0, \mathbf{s}, \mathbf{r}, (\mathbf{c}^{(\ell)}(s_0, (s_1, \dots, s_{w_1^{(\ell)}}), \hat{\mathbf{s}}^{(\ell)}, \mathbf{b}^{(\ell)}))_{\ell \in L_x}, \\ \mathbf{k}(-\sum_{\ell \in [L] \setminus \{\ell_y\}} \alpha^{(\ell)}, \mathbf{0}, \hat{\mathbf{r}}^{(\ell_y)}, 0) + \sum_{d \in [m_1^{(\ell_y)}]} \mathbf{k}_d^{(\ell_y)}(r_d, \mathbf{b}^{(\ell_y)} + \mathbf{b}_d^{(\ell_y)}), \\ (\mathbf{k}(\alpha^{(\ell)}, \mathbf{0}, \hat{\mathbf{r}}^{(\ell)}, 0) + \sum_{d \in [m_1^{(\ell)}]} \mathbf{k}_d^{(\ell)}(r_d, \mathbf{b}^{(\ell)} + \mathbf{b}_d^{(\ell)}))_{\ell \in [L] \setminus \{\ell_y\}} \end{array} \right\} \\
&\equiv \left\{ \begin{array}{c} s_0, \mathbf{s}, \mathbf{r}, (\mathbf{c}^{(\ell)}(s_0, (s_1, \dots, s_{w_1^{(\ell)}}), \hat{\mathbf{s}}^{(\ell)}, \mathbf{b}^{(\ell)}))_{\ell \in L_x}, \\ \mathbf{k}(-\sum_{\ell \in [L] \setminus \{\ell^\star\}} \alpha^{(\ell)}, \mathbf{0}, \hat{\mathbf{r}}^{(\ell^\star)}, 0) + \sum_{d \in [m_1^{(\ell^\star)}]} \mathbf{k}_d^{(\ell^\star)}(r_d, \mathbf{b}^{(\ell^\star)} + \mathbf{b}_d^{(\ell^\star)}), \\ (\mathbf{k}(\alpha^{(\ell)}, \mathbf{0}, \hat{\mathbf{r}}^{(\ell)}, 0) + \sum_{d \in [m_1^{(\ell)}]} \mathbf{k}_d^{(\ell)}(r_d, \mathbf{b}^{(\ell)}))_{\ell \in [L] \setminus \{\ell^\star\}} \end{array} \right\} \\
&= \left\{ \begin{array}{c} s_0, \mathbf{s}, \mathbf{r}, (\mathbf{c}^{(\ell)}(s_0, (s_1, \dots, s_{w_1^{(\ell)}}), \hat{\mathbf{s}}^{(\ell)}, \mathbf{b}^{(\ell)}))_{\ell \in L_x}, \\ \mathbf{k}(\mathbf{0}, \mathbf{0}, \hat{\mathbf{r}}^{(\ell^\star)}, 0) + \sum_{d \in [m_1^{(\ell^\star)}]} \mathbf{k}_d^{(\ell^\star)}(r_d, \mathbf{b}^{(\ell^\star)} + \mathbf{b}_d^{(\ell^\star)}) + \mathbf{k}(-\sum_{\ell \in [L] \setminus \{\ell^\star\}} \alpha^{(\ell)}, \mathbf{0}, \mathbf{0}, \mathbf{0}), \\ (\mathbf{k}(\alpha^{(\ell)}, \mathbf{0}, \hat{\mathbf{r}}^{(\ell)}, 0) + \sum_{d \in [m_1^{(\ell)}]} \mathbf{k}_d^{(\ell)}(r_d, \mathbf{b}^{(\ell)}))_{\ell \in [L] \setminus \{\ell^\star\}} \end{array} \right\}
\end{aligned}
$$

and

$$
\begin{aligned}
&\left\{ s_0, \mathbf{s}, \mathbf{r}, \mathbf{c}(s_0, \mathbf{s}, \hat{\mathbf{s}}, \mathbf{b}), \mathbf{k}(\alpha, \mathbf{0}, \hat{\mathbf{r}}, 0) + \sum_{d \in [m_1]} \mathbf{k}_d(r_d, \mathbf{b} + \mathbf{b}_d) \right\} \\
&= \left\{ \begin{array}{c} s_0, \mathbf{s}, \mathbf{r}, (\mathbf{c}^{(\ell)}(s_0, (s_1, \dots, s_{w_1^{(\ell)}}), \hat{\mathbf{s}}^{(\ell)}, \mathbf{b}^{(\ell)}))_{\ell \in L_x}, \\ \mathbf{k}(\alpha - \sum_{\ell \in [L] \setminus \{\ell_y\}} \alpha^{(\ell)}, \mathbf{0}, \hat{\mathbf{r}}^{(\ell_y)}, 0) + \sum_{d \in [m_1^{(\ell_y)}]} \mathbf{k}_d^{(\ell_y)}(r_d, \mathbf{b}^{(\ell_y)} + \mathbf{b}_d^{(\ell_y)}), \\ (\mathbf{k}(\alpha^{(\ell)}, \mathbf{0}, \hat{\mathbf{r}}^{(\ell)}, 0) + \sum_{d \in [m_1^{(\ell)}]} \mathbf{k}_d^{(\ell)}(r_d, \mathbf{b}^{(\ell)} + \mathbf{b}_d^{(\ell)}))_{\ell \in [L] \setminus \{\ell_y\}} \end{array} \right\} \\
&\equiv \left\{ \begin{array}{c} s_0, \mathbf{s}, \mathbf{r}, (\mathbf{c}^{(\ell)}(s_0, (s_1, \dots, s_{w_1^{(\ell)}}), \hat{\mathbf{s}}^{(\ell)}, \mathbf{b}^{(\ell)}))_{\ell \in L_x}, \\ \mathbf{k}(\alpha - \sum_{\ell \in [L] \setminus \{\ell^\star\}} \alpha^{(\ell)}, \mathbf{0}, \hat{\mathbf{r}}^{(\ell^\star)}, 0) + \sum_{d \in [m_1^{(\ell^\star)}]} \mathbf{k}_d^{(\ell^\star)}(r_d, \mathbf{b}^{(\ell^\star)} + \mathbf{b}_d^{(\ell^\star)}), \\ (\mathbf{k}(\alpha^{(\ell)}, \mathbf{0}, \hat{\mathbf{r}}^{(\ell)}, 0) + \sum_{d \in [m_1^{(\ell)}]} \mathbf{k}_d^{(\ell)}(r_d, \mathbf{b}^{(\ell)}))_{\ell \in [L] \setminus \{\ell^\star\}} \end{array} \right\} \\
&= \left\{ \begin{array}{c} s_0, \mathbf{s}, \mathbf{r}, (\mathbf{c}^{(\ell)}(s_0, (s_1, \dots, s_{w_1^{(\ell)}}), \hat{\mathbf{s}}^{(\ell)}, \mathbf{b}^{(\ell)}))_{\ell \in L_x}, \\ \mathbf{k}(\alpha, \mathbf{0}, \hat{\mathbf{r}}^{(\ell^\star)}, 0) + \sum_{d \in [m_1^{(\ell^\star)}]} \mathbf{k}_d^{(\ell^\star)}(r_d, \mathbf{b}^{(\ell^\star)} + \mathbf{b}_d^{(\ell^\star)}) + \mathbf{k}(-\sum_{\ell \in [L] \setminus \{\ell^\star\}} \alpha^{(\ell)}, \mathbf{0}, \mathbf{0}, \mathbf{0}), \\ (\mathbf{k}(\alpha^{(\ell)}, \mathbf{0}, \hat{\mathbf{r}}^{(\ell)}, 0) + \sum_{d \in [m_1^{(\ell)}]} \mathbf{k}_d^{(\ell)}(r_d, \mathbf{b}^{(\ell)}))_{\ell \in [L] \setminus \{\ell^\star\}} \end{array} \right\}
\end{aligned}
$$

where $s_0 \leftarrow_{\$} \mathbb{Z}_N$, $\mathbf{s} \leftarrow_{\$} \mathbb{Z}_N^{w_1}$, $\mathbf{r} \leftarrow_{\$} \mathbb{Z}_N^{m_1}$, $(\hat{\mathbf{s}}^{(\ell)})_{\ell \in L_x} \leftarrow_{\$} \mathbb{Z}_N^{w_2}$, $(\hat{\mathbf{r}}^{(\ell)})_{\ell \in L_y} \leftarrow_{\$} \mathbb{Z}_N^{m_2}$, $\mathbf{b} \leftarrow_{\$} \mathbb{Z}_N^n$, $(\alpha^{(\ell)})_{\ell \in [L_y]} \leftarrow_{\$} \mathbb{Z}_N^{|L_y|}$, $\alpha \leftarrow_{\$} \mathbb{Z}_N$, and $\mathbf{b}_d^{(\ell^\star)} \leftarrow \mathsf{Samp}(d, x^{(\ell^\star)}, y^{(\ell^\star)}, N)$ for $d \in [m_1^{(\ell^\star)}]$. Thus, to prove the relaxed perfect security (10), it is sufficient to show that

$$\left\{ \begin{array}{c} s_0, (s_1, \dots, s_{w_1^{(\ell^\star)}}), (r_1, \dots, r_{m_1^{(\ell^\star)}}), \mathbf{c}^{(\ell^\star)}(s_0, (s_1, \dots, s_{w_1^{(\ell^\star)}}), \hat{\mathbf{s}}^{(\ell^\star)}, \mathbf{b}^{(\ell^\star)}), \\ \mathbf{k}(0, \mathbf{0}, \hat{\mathbf{r}}^{(\ell^\star)}, \mathbf{0}) + \sum_{d \in [m_1^{(\ell^\star)}]} \mathbf{k}_d^{(\ell^\star)}(r_d, \mathbf{b}^{(\ell^\star)} + \mathbf{b}_d^{(\ell^\star)}) \end{array} \right\}$$
$$\approx \left\{ \begin{array}{c} s_0, (s_1, \dots, s_{w_1^{(\ell^\star)}}), (r_1, \dots, r_{m_1^{(\ell^\star)}}), \mathbf{c}^{(\ell^\star)}(s_0, (s_1, \dots, s_{w_1^{(\ell^\star)}}), \hat{\mathbf{s}}^{(\ell^\star)}, \mathbf{b}^{(\ell^\star)}), \\ \mathbf{k}(\alpha, \mathbf{0}, \hat{\mathbf{r}}^{(\ell^\star)}, \mathbf{0}) + \sum_{d \in [m_1^{(\ell^\star)}]} \mathbf{k}_d^{(\ell^\star)}(r_d, \mathbf{b}^{(\ell^\star)} + \mathbf{b}_d^{(\ell^\star)}) \end{array} \right\},$$

where $s_0 \leftarrow_{\$} \mathbb{Z}_N$, $(s_1, \dots, s_{w_1^{(\ell^\star)}}) \leftarrow_{\$} \mathbb{Z}_N^{w_1^{(\ell^\star)}}$, $(r_1, \dots, r_{m_1^{(\ell^\star)}}) \leftarrow_{\$} \mathbb{Z}_N^{m_1^{(\ell^\star)}}$, $\hat{\mathbf{s}}^{(\ell^\star)} \leftarrow_{\$} \mathbb{Z}_N^{w_2^{(\ell^\star)}}$, $\hat{\mathbf{r}}^{(\ell^\star)} \leftarrow_{\$} \mathbb{Z}_N^{m_2^{(\ell^\star)}}$, $\mathbf{b}^{(\ell^\star)} \leftarrow_{\$} \mathbb{Z}_N^{n^{(\ell^\star)}}$, $\alpha \leftarrow_{\$} \mathbb{Z}_N$, and $\mathbf{b}_d^{(\ell^\star)} \leftarrow \mathsf{Samp}(d, x^{(\ell^\star)}, y^{(\ell^\star)}, N)$ for $d \in [m_1^{(\ell^\star)}]$. Since the statistical indistinguishability is exactly the relaxed perfect security (10) of $\mathsf{PES}^{(\ell^\star)}$, the relaxed perfect security (10) of PES holds. □

Theorem 5.6. A $\mathsf{PES} = (\mathsf{Param}, \mathsf{EncC}, \mathsf{EncK}, \mathsf{Pair})$ for a predicate P described above satisfies the symbolic security when $\mathsf{PES}^{(\ell)}$ for $\ell \in [L]$ also satisfy the symbolic security and correctness. In particular, if $\mathsf{PES}^{(\ell)}$ for $\ell \in [L]$ satisfy $(d_1^{(\ell)}, d_2^{(\ell)})$-selective (resp. $(d_1^{(\ell)}, d_2^{(\ell)})$-co-selective) symbolic security, respectively, and correctness, PES satisfies $(\max_{\ell' \in [L]} d_1^{(\ell')}, \sum_{\ell' \in [L]} d_2^{(\ell')})$-selective (resp. $(\sum_{\ell' \in [L]} d_1^{(\ell')}, \sum_{\ell' \in [L]} d_2^{(\ell')})$-co-selective) symbolic security.

Proof. If $L_x \subset L_y$, PES satisfies the perfect security. Thus, PES also satisfies the symbolic security from Remark 7. Otherwise, i.e., $L_y \subseteq L_x$, we can prove that PES satisfies the symbolic security if there is an index $\ell^\star \in L_y$ such that $\mathsf{P}^{(\ell^\star)}(x^{(\ell^\star)}, y^{(\ell^\star)}) = 0$. For simplicity, we consider the case that there is only one such index satisfying $\ell^\star \neq \ell_y$. The proof for the other case is essentially the same. Hereafter, we use the fact that since $\mathsf{PES}^{(\ell)}$ for $\ell \in [L]$ satisfy $(d_1^{(\ell)}, d_2^{(\ell)})$-selective (resp. $(d_1^{(\ell)}, d_2^{(\ell)})$-co-selective) symbolic security, they also satisfy $(\max_{\ell' \in [L]} d_1^{(\ell')}, d_2^{(\ell)})$-selective (resp. $(\max_{\ell' \in [L]} d_1^{(\ell')}, d_2^{(\ell)})$-co-selective) symbolic security, respectively.

Selective Symbolic Security. At first, we describe $\mathsf{EncB}, \mathsf{EncS}$, and EncR for proving the selective symbolic security.

- $\mathsf{EncB}(x) \to (\mathbf{B}_1^{(\ell)}, \dots, \mathbf{B}_{n^{(\ell)}}^{(\ell)})_{\ell \in [L]}$: Set $\mathbf{B}_1^{(\ell)}, \dots, \mathbf{B}_{n^{(\ell)}}^{(\ell)}$ as uniformly random matrices if $\ell \notin L_x$. Otherwise, set $\mathbf{B}_1^{(\ell)}, \dots, \mathbf{B}_{n^{(\ell)}}^{(\ell)} \in \mathbb{Z}_N^{\max_{\ell' \in [L]} d_1^{(\ell')} \times \sum_{\ell' \in [L]} d_2^{(\ell')}}$ so that their left $\max_{\ell' \in [L]} d_1^{(\ell')} \times \sum_{\ell' \in [\ell-1]} d_2^{(\ell')}$ sub-matrices and right $\max_{\ell' \in [L]} d_1^{(\ell')} \times \sum_{\ell' \in [\ell+1, L]} d_2^{(\ell')}$ sub-matrices are zero matrices. Moreover, set the remaining $\max_{\ell' \in [L]} d_1^{(\ell')} \times d_2^{(\ell)}$ sub-matrices of them as corresponding outputs of $\mathsf{EncB}^{(\ell)}(x^{(\ell)})$.
- $\mathsf{EncR}(x, y) \to ((\mathbf{r}_1, \dots, \mathbf{r}_{m_1}), \mathbf{a}, (\mathbf{a}^{(\ell)})_{\ell \in L_y \setminus \{\ell_y\}}, (\hat{\mathbf{r}}_1^{(\ell)}, \dots, \hat{\mathbf{r}}_{m_2^{(\ell)}}^{(\ell)})_{\ell \in L_y})$: Find an index $\ell^\star \in L_x$ such that $\mathsf{P}^{(\ell^\star)}(x^{(\ell^\star)}, y^{(\ell^\star)}) = 0$ and proceed as follows.
  - Set $\mathbf{r}_1, \dots, \mathbf{r}_{m_1} \in \mathbb{Z}_N^{\max_{\ell' \in [L]} d_1^{(\ell')}}$ as corresponding outputs of $\mathsf{EncR}^{(\ell^\star)}(x^{(\ell^\star)}, y^{(\ell^\star)})$.

– Set $\mathbf{a} = \mathbf{a}^{(\ell^\star)} \in \mathbb{Z}_N^{\sum_{\ell' \in [L]} d_2^{(\ell')}}$ so that their left $\sum_{\ell' \in [\ell^\star - 1]} d_2^{(\ell')}$-dimensional sub-vector and right $\sum_{\ell' \in [\ell^\star + 1, L]} d_2^{(\ell')}$-dimensional sub-vector are zero vectors. Moreover, set the remaining $d_2^{(\ell^\star)}$-dimensional sub-vector as the corresponding output of $\mathsf{EncR}^{(\ell^\star)}(x^{(\ell^\star)}, y^{(\ell^\star)})$.
– Set all $(\mathbf{a}^{(\ell)})_{\ell \in L_y \setminus \{\ell_y, \ell^\star\}}$ as zero vectors. Thus, it holds that $\mathbf{a} - \sum_{\ell \in L_y \setminus \{\ell_y\}} \mathbf{a}^{(\ell)} = \mathbf{0}$.
– For $\ell \in L_y$, set $\hat{\mathbf{r}}_1^{(\ell)}, \ldots, \hat{\mathbf{r}}_{m_2^{(\ell)}}^{(\ell)} \in \mathbb{Z}_N^{\sum_{\ell' \in [L]} d_2^{(\ell')}}$ so that their left $\sum_{\ell' \in [\ell - 1]} d_2^{(\ell')}$-dimensional sub-vectors and right $\sum_{\ell' \in [\ell + 1, L]} d_2^{(\ell')}$-dimensional sub-vectors are zero vectors. Moreover, set the remaining $d_2^{(\ell)}$-dimensional sub-vectors as the corresponding outputs of $\mathsf{EncR}^{(\ell)}(x^{(\ell)}, y^{(\ell)})$.

- $\mathsf{EncS}(x) \to (\mathbf{s}_0, (\mathbf{s}_1, \ldots, \mathbf{s}_{w_1}), (\hat{\mathbf{s}}_1^{(\ell)}, \ldots, \hat{\mathbf{s}}_{w_2^{(\ell)}}^{(\ell)})_{\ell \in L_x})$: Proceed as follows.
  – Set $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{w_1} \in \mathbb{Z}_N^{\sum_{\ell' \in [L]} d_2^{(\ell')}}$ by $\sum_{\ell \in L_x} \mathbf{s}_0^{(\ell)}, \sum_{\ell \in L_x} \mathbf{s}_1^{(\ell)}, \ldots, \sum_{\ell \in L_x} \mathbf{s}_{w_1}^{(\ell)}$, where left $\sum_{\ell' \in [\ell - 1]} d_2^{(\ell')}$-dimensional sub-vector and right $\sum_{\ell' \in [\ell + 1, L]} d_2^{(\ell')}$-dimensional sub-vector of $\mathbf{s}_0^{(\ell)}, \mathbf{s}_1^{(\ell)}, \ldots, \mathbf{s}_{w_1}^{(\ell)}$ are zero vectors. Moreover, set the remaining $d_2^{(\ell)}$-dimensional sub-vector as the corresponding output of $\mathsf{EncS}^{(\ell)}(x^{(\ell)})$.
  – Set $\hat{\mathbf{s}}_1^{(\ell)}, \ldots, \hat{\mathbf{s}}_{w_2^{(\ell)}}^{(\ell)} \in \mathbb{Z}_N^{\max_{\ell' \in [L]} d_1^{(\ell')}}$ as the corresponding outputs of $\mathsf{EncS}^{(\ell)}(x^{(\ell)})$.

We show that the above $\mathsf{EncB}, \mathsf{EncR}$, and EncS satisfy all the requirements of selective symbolic security in Definition 5.3. By construction, it holds that $\langle \mathbf{s}_0, \mathbf{a} \rangle = \langle \mathbf{s}_0, \mathbf{a}^{(\ell^\star)} \rangle$. Since $\mathsf{P}^{(\ell^\star)}(x^{(\ell^\star)}, y^{(\ell^\star)}) = 0$ holds, the selective symbolic security of $\mathsf{PES}^{(\ell^\star)}$ ensures that $\langle \mathbf{s}_0, \mathbf{a} \rangle \neq 0$. Next, by substituting the outputs of $\mathsf{EncB}, \mathsf{EncR}$, and EncS to the variables of PES, an evaluation result is the same as a sum of evaluation results for $(\mathsf{PES}^{(\ell)})_{\ell \in L_y}$. By construction, the variable α for $\mathsf{PES}^{(\ell)}$ is substituted as $\mathbf{a}^{(\ell^\star)}$ for $\ell = \ell^\star$ and zero vectors otherwise. Since $\mathsf{PES}^{(\ell)}(x^{(\ell)}, y^{(\ell)}) = 1$ holds for $\ell \in L_y \setminus \{\ell^\star\}$, the correctness of $(\mathsf{PES}^{(\ell)})_{\ell \in L_y \setminus \{\ell^\star\}}$ ensures that evaluation results for them are $\langle \mathbf{s}_0, \mathbf{a}^{(\ell)} \rangle = \langle \mathbf{s}_0, \mathbf{0} \rangle = 0$ and the selective symbolic security for $\mathsf{PES}^{(\ell^\star)}$ ensures that the evaluation result for $\mathsf{PES}^{(\ell^\star)}$ is 0. Therefore, the evaluation result for PES is also 0. Thus, we complete the proof of selective symbolic security.

co-Selective Symbolic Security. Next, we describe $\mathsf{EncB}, \mathsf{EncS}$, and EncR for proving the co-selective symbolic security.

- $\mathsf{EncB}(y) \to (\mathbf{B}_1^{(\ell)}, \ldots, \mathbf{B}_{n^{(\ell)}}^{(\ell)})_{\ell \in [L]}$: Set $\mathbf{B}_1^{(\ell)}, \ldots, \mathbf{B}_{n^{(\ell)}}^{(\ell)}$ as uniformly random matrices if $\ell \notin L_y$. Otherwise, set $\mathbf{B}_1^{(\ell)}, \ldots, \mathbf{B}_{n^{(\ell)}}^{(\ell)} \in \mathbb{Z}_N^{\sum_{\ell' \in [L]} d_1^{(\ell')} \times \sum_{\ell' \in [L]} d_2^{(\ell')}}$ so that their left $\sum_{\ell' \in [L]} d_1^{(\ell')} \times \sum_{\ell' \in [\ell - 1]} d_2^{(\ell')}$ sub-matrices and right $\sum_{\ell' \in [L]} d_1^{(\ell')} \times \sum_{\ell' \in [\ell + 1, L]} d_2^{(\ell')}$ sub-matrices are zero matrices. Moreover, set the remaining $\sum_{\ell' \in [L]} d_1^{(\ell')} \times d_2^{(\ell)}$ sub-matrices so that their top $\sum_{\ell' \in [\ell - 1]} d_1^{(\ell')} \times d_2^{(\ell)}$ sub-matrices and bottom $\sum_{\ell' \in [\ell + 1, L]} d_1^{(\ell')} \times d_2^{(\ell)}$ sub-matrices are zero matrices. Finally, set the remaining $d_1^{(\ell)} \times d_2^{(\ell)}$ sub-matrices as the corresponding outputs of $\mathsf{EncB}^{(\ell)}(y^{(\ell)})$.

- $\mathsf{EncR}(y) \to ((\mathbf{r}_1, \ldots, \mathbf{r}_{m_1}), \mathbf{a}, (\mathbf{a}^{(\ell)})_{\ell \in L_y \setminus \{\ell_y\}}, (\hat{\mathbf{r}}_1^{(\ell)}, \ldots, \hat{\mathbf{r}}_{m_2^{(\ell)}}^{(\ell)})_{\ell \in L_y})$: Proceed as follows.

- Set $\mathbf{r}_1, \dots, \mathbf{r}_{m_1} \in \mathbb{Z}_N^{\sum_{\ell' \in [L]} d_1^{(\ell')}}$ so that their left $\sum_{\ell' \in [\ell-1]} d_1^{(\ell')}$-dimensional sub-vectors and right $\sum_{\ell' \in [\ell+1, L]} d_1^{(\ell')}$-dimensional sub-vectors are zero vectors. Moreover, set the remaining $d_1^{(\ell)}$-dimensional sub-vectors as the corresponding outputs of $\mathsf{EncR}^{(\ell)}(y^{(\ell)})$.
- Set $\mathbf{a} = \sum_{\ell \in L_y} \mathbf{a}^{(\ell)} \in \mathbb{Z}_N^{\sum_{\ell' \in [L]} d_2^{(\ell')}}$, where all $\mathbf{a}^{(\ell)}$ are defined below.
- Set $\mathbf{a}^{(\ell)} \in \mathbb{Z}_N^{\sum_{\ell' \in [L]} d_2^{(\ell')}}$ so that their left $\sum_{\ell' \in [\ell-1]} d_2^{(\ell')}$-dimensional sub-vector and right $\sum_{\ell' \in [\ell+1, L]} d_2^{(\ell')}$-dimensional sub-vector are zero vectors. Moreover, set the remaining $d_2^{(\ell)}$-dimensional sub-vector as the corresponding output of $\mathsf{EncR}^{(\ell)}(y^{(\ell)})$.
- For $\ell \in L_y$, set $\hat{\mathbf{r}}_1^{(\ell)}, \dots, \hat{\mathbf{r}}_{m_2^{(\ell)}}^{(\ell)} \in \mathbb{Z}_N^{\sum_{\ell' \in [L]} d_2^{(\ell')}}$ so that their left $\sum_{\ell' \in [\ell-1]} d_2^{(\ell')}$-dimensional sub-vectors and right $\sum_{\ell' \in [\ell+1, L]} d_2^{(\ell')}$-dimensional sub-vectors are zero vectors. Moreover, set the remaining $d_2^{(\ell)}$-dimensional sub-vectors as the corresponding outputs of $\mathsf{EncR}^{(\ell)}(y^{(\ell)})$.

- $\mathsf{EncS}(x, y) \to (\mathbf{s}_0, (\mathbf{s}_1, \dots, \mathbf{s}_{w_1}), (\hat{\mathbf{s}}_1^{(\ell)}, \dots, \hat{\mathbf{s}}_{w_2^{(\ell)}}^{(\ell)})_{\ell \in L_x})$: Find an index $\ell^\star \in L_x$ such that $\mathsf{P}^{(\ell^\star)}(x^{(\ell^\star)}, y^{(\ell^\star)}) = 0$ and proceed as follows.
  - Set $\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{w_1} \in \mathbb{Z}_N^{\sum_{\ell' \in [L]} d_2^{(\ell')}}$ so that their left $\sum_{\ell' \in [\ell^\star - 1]} d_2^{(\ell')}$-dimensional sub-vector and right $\sum_{\ell' \in [\ell^\star + 1, L]} d_2^{(\ell')}$-dimensional sub-vector are zero vectors. Moreover, set the remaining $d_2^{(\ell^\star)}$-dimensional sub-vector as the corresponding output of $\mathsf{EncS}^{(\ell^\star)}(x^{(\ell^\star)}, y^{(\ell^\star)})$.
  - For $\ell \in L_x$, set $\hat{\mathbf{s}}_1^{(\ell)}, \dots, \hat{\mathbf{s}}_{w_2^{(\ell)}}^{(\ell)} \in \mathbb{Z}_N^{\sum_{\ell' \in [L]} d_1^{(\ell')}}$ so that their left $\sum_{\ell' \in [\ell-1]} d_1^{(\ell')}$-dimensional sub-vectors and right $\sum_{\ell' \in [\ell+1, L]} d_1^{(\ell')}$-dimensional sub-vectors are zero vectors. Moreover, set the remaining $d_1^{(\ell)}$-dimensional sub-vectors as the corresponding outputs of $\mathsf{EncS}^{(\ell)}(x^{(\ell)}, y^{(\ell)})$.

We show that the above $\mathsf{EncB}, \mathsf{EncR}$, and EncS satisfy all the requirements of co-selective symbolic security in Definition 5.3. By construction, it holds that $\langle \mathbf{s}_0, \mathbf{a} \rangle = \langle \mathbf{s}_0, \mathbf{a}^{(\ell^\star)} \rangle$. Since $\mathsf{P}^{(\ell^\star)}(x^{(\ell^\star)}, y^{(\ell^\star)}) = 0$ holds, the co-selective symbolic security of $\mathsf{PES}^{(\ell^\star)}$ ensures that $\langle \mathbf{s}_0, \mathbf{a} \rangle \neq 0$. Next, by substituting the outputs of $\mathsf{EncB}, \mathsf{EncR}$, and EncS to the variables of PES, an evaluation result is the same as a sum of evaluation results for $(\mathsf{PES}^{(\ell)})_{\ell \in L_y}$. Since $\mathsf{PES}^{(\ell)}(x^{(\ell)}, y^{(\ell)}) = 1$ holds for $\ell \in L_y \setminus \{\ell^\star\}$, the correctness of $(\mathsf{PES}^{(\ell)})_{\ell \in L_y \setminus \{\ell^\star\}}$ ensures that evaluation results for them are $\langle \mathbf{s}_0, \mathbf{a}^{(\ell)} \rangle = 0$ and the co-selective symbolic security for $\mathsf{PES}^{(\ell^\star)}$ ensures that the evaluation result for $\mathsf{PES}^{(\ell^\star)}$ is 0. Therefore, the evaluation result for PES is also 0. Thus, we complete the proof of co-selective symbolic security. □

5.3 Proposed Scheme for KP-ABE

At first, we review non-monotone span programs. Let $\mathbb{Z}_p$ denote a universe of attributes with an exponentially large prime p. A span program is a linear secret sharing scheme (A, π), where $A \in \mathbb{Z}_p^{n_1 \times n_2}$ is a matrix whose i-th row is denoted by A_i and $\pi : [n_1] \to \{0, 1\} \times \mathbb{Z}_p$ is a map. In the

case of monotone span programs (MSP), the first output of π is always 0. When a set of attributes $S \subset \mathbb{Z}_p$ is given access to a span program (A, π), we define a map $\gamma : [n_1] \to \{0,1\}$ for (A, π) such that $\gamma(i) = 1$ iff $(\pi(i) = (0, s) \wedge s \in S) \vee (\pi(i) = (1, s) \wedge s \notin S)$. To specify the set of attributes S explicitly, we may also use a notation $\gamma(i, S)$. Let $A_S := \{A_i : i \in [n_1] \wedge \gamma(i, S) = 1\}$ be a matrix whose rows consist of a subset of A. An access structure (A, π) is said to accept a set of attributes S iff $\vec{1} \in \mathsf{Span}(A_S)$ holds, where $\vec{1} := (1, 0, \ldots, 0) \in \mathbb{Z}_p^{n_2}$. Otherwise, (A, π) is said to reject S. If S is accepted, we can efficiently compute a set of integers c_i's such that $\sum_{i:A_i \in A_S} c_i A_i = \vec{1}$. In contrast, it is known [Bei11] that if (A, π) rejects S, there is a column vector $\mathbf{v} = (v_1, \ldots, v_{n_2})^\top$ such that $v_1 = 1$ and $A_i \mathbf{v} = 0$ for all i such that $\gamma(i, S) = 0$. Let I and $\bar{I}$ denote sets of indices such that $I := \{i : i \in [n_1] \wedge \pi(i) = (0, *)\}$ and $\bar{I} := \{i : i \in [n_1] \wedge \pi(i) = (1, *)\}$, where $I \cup \bar{I} = [n_1]$ and $I \cap \bar{I} = \emptyset$ hold.

Then, we propose a PES for KP-ABE for non-monotone span programs with compact ciphertexts.

$\mathsf{Param}(\mathsf{par}) \to T + 1$: Let $\mathbf{b} := (b_t)_{t \in [0,T]}$.

$\mathsf{EncC}(S, N) \to c$:

$$c := s(w_0 b_0 + \cdots + w_T b_T),$$

where $\mathbf{s} := s$, and w_j is a coefficient of x^j in $q(x) := \prod_{y \in S}(x - y)$.

$\mathsf{EncK}((A, \pi), T + 1) \to \mathbf{k} := ((k_{1,i})_{i \in [n_1]}, (k_{2,i,t})_{i \in I, t \in [T]}, (\bar{k}_{2,i,t})_{i \in \bar{I}, t \in [0,T]})$:

$$\begin{aligned}
&k_{1,i} := A_i(\alpha, v_2, \ldots, v_{n_2})^\top + \phi_i, \\
&k_{2,i,1} := \phi_i + r_i(b_1 - \pi(i) b_0), \qquad k_{2,i,t} := r_i(b_t - \pi(i)^t b_0) \quad \text{for } t \in [2, T], \\
&\bar{k}_{2,i,t} := \pi(i)^t \phi_i + r_i b_t \quad \text{for } t \in [0, T],
\end{aligned}$$

where $\mathbf{r} := (r_i)_{i \in [n_1]}$, and $\hat{\mathbf{r}} := (v_2, \ldots, v_{n_2}, (\phi_i)_{i \in [n_1]})$.

Correctness. Here, we informally explain how to recover $sA_i(\alpha, v_2, \ldots, v_{n_2})^\top$ for all $i \in [n_1]$ such that $\gamma(i) = 1$ since they are sufficient to recover αs. For a fixed $i^\star \in [n_1]$, it suffices to compute $s\phi_{i^\star}$ for this purpose. In other words, we can recover αs from

$$sk_{1,i} - s\phi_{i^\star} = sA_i(\alpha, v_2, \ldots, v_{n_2})^\top$$

for all $i \in [n_1]$ such that $\gamma(i) = 1$.

If $i^\star \in I$, by taking a linear combination of $k_{2,i^\star,1}, \ldots, k_{2,i^\star,T}$ with $w_1, w_2, \ldots, w_T$, we have

$$\begin{aligned}
&w_1 k_{2,i^\star,1} + \cdots + w_T k_{2,i^\star,T} \\
&= w_1 \phi_{i^\star} + \sum_{t \in [T]} w_t (r_{i^\star}(b_t - \pi(i^\star)^t b_0)) \\
&= w_1 \phi_{i^\star} + r_{i^\star} \left(w_1 b_1 + \cdots + w_T b_T - (\pi(i^\star) w_1 + \cdots + \pi(i^\star)^T w_T) b_0\right) \\
&= w_1 \phi_{i^\star} + r_{i^\star} \left(w_1 b_1 + \cdots + w_T b_T - (p(\pi(i^\star)) - w_0) b_0\right) \\
&= w_1 \phi_{i^\star} + r_{i^\star}(w_0 b_0 + w_1 b_1 + \cdots + w_T b_T).
\end{aligned}$$

Here, we use the fact that $q(\pi(i^\star)) = 0$ since $\pi(i^\star) \in S \Leftrightarrow \gamma(i^\star) = 1 \wedge i^\star \in I$. Thus, we have $s\phi_{i^\star}$ by computing

$$\frac{1}{w_1} \cdot \left(s(w_1 k_{2,i^\star,1} + \cdots + w_T k_{2,i^\star,T}) - r_{i^\star} c\right) = s\phi_{i^\star}.$$

Next, we show how to recover $s\phi_{i^\star}$ for a fixed $i^\star \in \bar{I}$. By taking a linear combination of $\bar{k}_{2,i^\star,0}, \ldots, \bar{k}_{2,i^\star,T}$ with $w_0, \ldots, w_T$, we have

$$\begin{aligned} w_0\bar{k}_{2,i^\star,0} + \cdots + w_T\bar{k}_{2,i^\star,T} &= \sum_{t\in[0,T]} w_t(\pi(i^\star)^t\phi_{i^\star} + r_{i^\star}b_t) \\ &= q(\pi(i^\star))\phi_{i^\star} + r_{i^\star}(w_0b_0 + \cdots + w_Tb_T), \end{aligned}$$

where $q(\pi(i^\star)) \neq 0$ since $\pi(i^\star) \notin S \Leftrightarrow \gamma(i^\star) = 1 \wedge i^\star \in \bar{I}$. Thus, we have $s\phi_{i^\star}$ by computing

$$\frac{1}{q(\pi(i^\star))} \cdot s\left((w_0\bar{k}_{2,i^\star,0} + \cdots + w_T\bar{k}_{2,i^\star,T}) - r_{i^\star}c\right) = s\phi_{i^\star}.$$

Relaxed Perfect Security. We prove the relaxed perfect security of the proposed PES.

Lemma 5.1 (Relaxed Perfect Security of the PES)**.** The above PES of KP-ABE for non-monotone span programs with compact ciphertexts satisfies the relaxed perfect security.

Proof. We define the outputs $(z_{d,0}, \ldots, z_{d,T}) \leftarrow_{\$} \mathsf{Samp}(d, x, y, N)$ as follows:

- If $\gamma(d, S) = 1$, $(z_{d,0}, \ldots, z_{d,T})$ is a zero vector.
- If $\gamma(d, S) = 0 \wedge d \in I$, $(z_{d,0}, \ldots, z_{d,T})$ is a uniformly random vector.
- If $\gamma(d, S) = 0 \wedge d \in \bar{I}$, $(z_{d,0}, \ldots, z_{d,T}) = (\phi'_d, \phi'_d \cdot \pi(d), \ldots, \phi'_d \cdot \pi^T(d))$, where $\phi'_d \leftarrow_{\$} \mathbb{Z}_N$.

If $\gamma(d, S) = 1$, the left and right distributions of (9) are the same. If $\gamma(d, S) = 0 \wedge d \in I$, the left distribution of (9) is given by

$$\left\{s, r_d, s(w_0b_0 + \cdots + w_Tb_T), (r_d(b_t - \pi(d)^t b_0))_{t\in[T]}\right\}.$$

We can specify the distribution by

$$\begin{bmatrix} w_0 & w_1 & \cdots & w_T \\ -\pi(\tau) & 1 & & \\ \vdots & & \ddots & \\ -\pi(\tau)^T & & & 1 \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_T \end{bmatrix}, \tag{11}$$

where $(b_0, \ldots, b_T) \leftarrow_{\$} \mathbb{Z}_N^{T+1}$. Since $\pi(d) \notin S$, the left matrix of (11) is non-singular. Thus, the value of (11) is uniformly random in $\mathbb{Z}_N^{T+1}$. Therefore, the left and right distributions of (9) are statistically indistinguishable.

If $\gamma(d, S) = 0 \wedge d \in \bar{I}$, the left distribution of (9) is given by

$$\left\{s, r_d, s(w_0b_0 + \cdots + w_Tb_T), (r_db_t)_{t\in[0,T]}\right\}.$$

We can specify the distribution by

$$\begin{bmatrix} w_0 & w_1 & \cdots & w_T \\ 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_T \end{bmatrix}, \tag{12}$$

where $(b_0, \ldots, b_T) \leftarrow_{\$} \mathbb{Z}_N^{T+1}$. Since the bottom $(T+1) \times (T+1)$ submatrix of the left matrix of (12) is an identity matrix, the only way to check the distribution is whether multiplying a row vector $(-1, w_0, w_1, \ldots, w_T)$ from the left becomes a zero vector or not. On the other hand, the right distribution of (9) is given by

$$\Big\{s, r_d, s(w_0 b_0 + \cdots + w_T b_T), (r_d(b_t + \phi'_d \cdot \pi(d)^t)_{t\in[0,T]}\Big\}.$$

We can specify the distribution by

$$\begin{bmatrix} w_0 & w_1 & \cdots & w_T \\ 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_T \end{bmatrix} + \phi'_d \cdot \begin{bmatrix} 0 \\ 1 \\ \pi(d) \\ \vdots \\ \pi(d)^T \end{bmatrix}, \tag{13}$$

where $(b_0, \ldots, b_T) \leftarrow_{\$} \mathbb{Z}_N^{T+1}$ and $\phi'_d \leftarrow_{\$} \mathbb{Z}_N$. Since $\pi(d) \in S$, multiplying a row vector $(-1, w_0, w_1, \ldots, w_T)$ from the left of (13) becomes a zero vector. Therefore, the left and right distributions of (9) are statistically indistinguishable.

Finally, we prove the second indistinguishability (10). The only difference between the two distribution is that $(0, v_2, \ldots, v_{n_2})^\top$ in $k_{1,i}$ of the left distribution are replaced by $(\alpha, v_2, \ldots, v_{n_2})^\top$, where $\alpha \leftarrow_{\$} \mathbb{Z}_N$. If $\gamma(i) = 0 \wedge i \in I$, $k_{2,i,1}$ is uniformly random in $\mathbb{Z}_N$. Since the information of ϕ_i disappears, $k_{1,i}$ is also uniformly random in $\mathbb{Z}_N$. If $\gamma(i) = 0 \wedge i \in \bar{I}$, $\bar{k}_{2,i,1}$ distributes according to $\pi(i)^t(\phi_i + \phi'_i) + r_i b_t$, where $\phi'_i \leftarrow_{\$} \mathbb{Z}_N$. Since the information of ϕ_i is masked by ϕ'_i, $k_{1,i}$ is also uniformly random in $\mathbb{Z}_N$. Thus, what we have to show is that

$$\{A_i \cdot (0, v_2, \ldots, v_{n_2})^\top\}_{i:\gamma(i)=1} \approx \{A_i \cdot (\alpha, v_2, \ldots, v_{n_2})^\top\}_{i:\gamma(i)=1}. \tag{14}$$

Let $\mathbf{a}^\perp \in \mathbb{Z}_N^{n_2}$ denote a vector whose first element is 1 and satisfying $A_i \mathbf{a}^\perp = \mathbf{0}$ for all i such that $\gamma(i) = 1$. Here, we replace $A_i \cdot (0, v_2, \ldots, v_{n_2})^\top$ which is the left distribution of (14) by $A_i \cdot \big((0, v_2, \ldots, v_{n_2})^\top + \alpha \mathbf{a}^\perp\big)$, where $v_2, \ldots, v_{n_2}, \alpha \leftarrow_{\$} \mathbb{Z}_N$. The modification does not change the distribution since

$$\begin{aligned} A_i \cdot \big((0, v_2, \ldots, v_{n_2})^\top + \alpha \mathbf{a}^\perp\big) &= A_i \cdot (0, v_2, \ldots, v_{n_2})^\top + \alpha A_i \mathbf{a}^\perp \\ &= A_i \cdot (0, v_2, \ldots, v_{n_2})^\top. \end{aligned}$$

Furthermore, $(0, v_2, \ldots, v_{n_2})^\top + \alpha \mathbf{a}^\perp = (\alpha, v_2 + \alpha a_2^\perp, \ldots, v_{n_2} + \alpha a_{n_2}^\perp)^\top$ holds, where $a_j^\perp$ denotes the j-th elements of $\mathbf{a}^\perp$. Since all $\alpha, v_2 + \alpha a_2^\perp, \ldots, v_{n_2} + \alpha a_{n_2}^\perp$ distribute uniformly in $\mathbb{Z}_N$, they follow according to the right distribution of (14). Thus, we complete the proof. □

Comparison. Our proposed PES for *non-monotone* span programs have a $T+1$ common variable, one ciphertext-encoding polynomial with one non-lone ciphertext encoding variable, and $O(n_1 T)$ key-encoding polynomials with n_1 key-encoding variables. In contrast, Agrawal and Chase's PES for *monotone* span programs have $T+6$ common variable, three ciphertext-encoding polynomials with three ciphertext-encoding variables, and $O(n_1 T)$ key-encoding polynomials with $O(n_1 + n_2)$ key-encoding variables. Thus, although the proposed PES supports more complex non-monotone predicate, it is more efficient than Agrawal and Chase's one.

6 Conclusion

In this paper, we proposed a generic construction of CCA-secure ABEET from IND-CPA-secure delegatable ABE with the hierarchical depth three. The construction is an attribute-based extension of Lee et al.'s generic construction of CCA-secure IBEET from IND-CPA-secure hierarchical IBE with the depth three [LLS+16b]. To achieve CCA security, we used Yamada et al.'s technique [YAH+11]. Based on the predicate encoding and pair encoding frameworks [Att14, Wee14] and known lattice-based delegatable ABE schemes [ACM12, Xag13, BGG+14], we obtain various ABEET schemes with new properties that have not been achieved so far. However, since there are no generic methods for non-delegatable ABE to satisfy the delegatability, there are several open questions. Although we obtained ABEET schemes for (non-)monotone span programs (Schemes 1–12) from ABE schemes for the same predicates in the standard model, there are more efficient schemes in the random oracle model [AC17a, TKN20]. Although we obtained the first ABEET schemes for deterministic finite automata (Schemes 13 and 14) under the q-ratio assumption, there are ABE schemes for the same predicate under the standard k-linear assumption [AMY19b, GWW19, GW20] and ABE schemes for non-deterministic finite automata under the LWE assumptions [AMY19a]. Although we obtained selectively secure lattice-based ABEET schemes for circuits and inner-product predicates, there are semi-adaptively secure lattice-based ABE scheme for circuits [BV16] and adaptively secure lattice-based inner-product encryption [KNY+20]. Therefore, it is an interesting open problem to construct CCA-secure ABEET schemes with these properties. In addition to the construction of ABEET, we proposed a delegatable transformation of pair encoding and a new pair encoding scheme of key-policy ABE for non-monotone span programs with compact ciphertexts.

Acknowledgments

This work is supported by JSPS KAKENHI Grant Numbers JP18H05289, JP18K11293, JP21H03441, JP23K21668, JP23KJ0968, and JP24K02939, and MEXT Leading Initiative for Excellent Young Researchers.

References

[ABC+08] Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. "Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions." In: *J. Cryptol.* 21.3 (2008), pp. 350–391.

[ABS17] Miguel Ambrona, Gilles Barthe, and Benedikt Schmidt. "Generic Transformations of Predicate Encodings: Constructions and Applications." In: *CRYPTO*. 2017, pp. 36–66.

[AC16a] Shashank Agrawal and Melissa Chase. "A Study of Pair Encodings: Predicate Encryption in Prime Order Groups." In: *TCC*. 2016, pp. 259–288.

[AC16b] Shashank Agrawal and Melissa Chase. "A Study of Pair Encodings: Predicate Encryption in Prime Order Groups." In: *TCC*. 2016, pp. 259–288.

[AC17a] Shashank Agrawal and Melissa Chase. "FAME: Fast Attribute-based Message Encryption." In: *ACM CCS*. 2017, pp. 665–682.

[AC17b] Shashank Agrawal and Melissa Chase. "Simplifying Design and Analysis of Complex Predicate Encryption Schemes." In: *EUROCRYPT*. 2017, pp. 627–656.

[ACM12] Michel Abdalla, Angelo De Caro, and Karina Mochetti. "Lattice-Based Hierarchical Inner Product Encryption." In: *LATINCRYPT*. 2012, pp. 121–138.

[AET+22] Kyoichi Asano, Keita Emura, Atsushi Takayasu, and Yohei Watanabe. "A Generic Construction of CCA-secure Attribute-based Encryption with Equality Test." In: *ProvSec*. 2022, pp. 3–19.

[AET24] Kyoichi Asano, Keita Emura, and Atsushi Takayasu. "More Efficient Adaptively Secure Lattice-Based IBE with Equality Test in the Standard Model." In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E107.A.3 (2024), pp. 248–259.

[AMY19a] Shweta Agrawal, Monosij Maitra, and Shota Yamada. "Attribute Based Encryption (and more) for Nondeterministic Finite Automata from LWE." In: *CRYPTO*. 2019, pp. 765–797.

[AMY19b] Shweta Agrawal, Monosij Maitra, and Shota Yamada. "Attribute Based Encryption for Deterministic Finite Automata from DLIN." In: *TCC*. 2019, pp. 91–117.

[Att14] Nuttapong Attrapadung. "Dual System Encryption via Doubly Selective Security: Framework, Fully Secure Functional Encryption for Regular Languages, and More." In: *EUROCRYPT*. 2014, pp. 557–577.

[Att16] Nuttapong Attrapadung. "Dual System Encryption Framework in Prime-Order Groups via Computational Pair Encodings." In: *ASIACRYPT*. 2016, pp. 591–623.

[Att19] Nuttapong Attrapadung. "Unbounded Dynamic Predicate Compositions in Attribute-Based Encryption." In: *EUROCRYPT*. 2019, pp. 34–67.

[BCO+04] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. "Public Key Encryption with Keyword Search." In: *EUROCRYPT*. 2004, pp. 506–522.

[Bei11] Amos Beimel. "Secret-Sharing Schemes: A Survey." In: *IWCC*. 2011, pp. 11–46.

[BGG+14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. "Fully Key-Homomorphic Encryption, Arithmetic Circuit ABE and Compact Garbled Circuits." In: *EUROCRYPT*. 2014, pp. 533–556.

[BV16] Zvika Brakerski and Vinod Vaikuntanathan. "Circuit-ABE from LWE: Unbounded Attributes and Semi-adaptive Security." In: *CRYPTO*. 2016, pp. 363–384.

[CG17] Jie Chen and Junqing Gong. "ABE with Tag Made Easy - Concise Framework and New Instantiations in Prime-Order Groups." In: *ASIACRYPT*. 2017, pp. 35–65.

[CGW15] Jie Chen, Romain Gay, and Hoeteck Wee. "Improved Dual System ABE in Prime-Order Groups via Predicate Encodings." In: *EUROCRYPT*. 2015, pp. 595–624.

[CHH+18] Yuzhao Cui, Qiong Huang, Jianye Huang, Hongbo Li, and Guomin Yang. "Outsourced Ciphertext-Policy Attribute-Based Encryption with Equality Test." In: *Inscrypt*. 2018, pp. 448–467.

[CHH+19] Yuzhao Cui, Qiong Huang, Jianye Huang, Hongbo Li, and Guomin Yang. "Ciphertext-Policy Attribute-Based Encrypted Data Equality Test and Classification." In: *Comput. J.* 62.8 (2019), pp. 1166–1177.

[CHK04] Ran Canetti, Shai Halevi, and Jonathan Katz. "Chosen-Ciphertext Security from Identity-Based Encryption." In: *EUROCRYPT*. 2004, pp. 207–222.

[DFK+19] Dung Hoang Duong, Kazuhide Fukushima, Shinsaku Kiyomoto, Partha Sarathi Roy, and Willy Susilo. "A Lattice-Based Public Key Encryption with Equality Test in Standard Model." In: *ACISP*. 2019, pp. 138–155.

[DLR+19] Dung Hoang Duong, Huy Quoc Le, Partha Sarathi Roy, and Willy Susilo. "Lattice-Based IBE with Equality Test in Standard Model." In: *ProvSec*. 2019, pp. 19–40.

[DSB+19] Dung Hoang Duong, Willy Susilo, Minh Kim Bui, and Thanh Xuan Khuc. "A Lattice-Based Certificateless Public Key Encryption with Equality Test in Standard Model." In: *Inscrypt*. 2019, pp. 50–65.

[GW20] Junqing Gong and Hoeteck Wee. "Adaptively Secure ABE for DFA from k-Lin and More." In: *EUROCRYPT*. 2020, pp. 278–308.

[GWW19] Junqing Gong, Brent Waters, and Hoeteck Wee. "ABE for DFA from k-Lin." In: *CRYPTO*. 2019, pp. 732–764.

[HTC+14] Kaibin Huang, Raylin Tso, Yu-Chi Chen, Wangyu Li, and Hung-Min Sun. "A New Public Key Encryption with Equality Test." In: *NSS*. 2014, pp. 550–557.

[HTC+15] Kaibin Huang, Raylin Tso, Yu-Chi Chen, Sk. Md. Mizanur Rahman, Ahmad Almogren, and Atif Alamri. "PKE-AET: Public Key Encryption with Authorized Equality Test." In: *Comput. J.* 58.10 (2015), pp. 2686–2697.

[KNY+20] Shuichi Katsumata, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. "Adaptively Secure Inner Product Encryption from LWE." In: *ASIACRYPT*. 2020, pp. 375–404.

[LLS+16a] Hyung Tae Lee, San Ling, Jae Hong Seo, and Huaxiong Wang. "CCA2 Attack and Modification of Huang *et al.*'s Public Key Encryption with Authorized Equality Test." In: *Comput. J.* 59.11 (2016), pp. 1689–1694.

[LLS+16b] Hyung Tae Lee, San Ling, Jae Hong Seo, and Huaxiong Wang. "Semi-generic construction of public key encryption and identity-based encryption with equality test." In: *Inf. Sci.* 373 (2016), pp. 419–440.

[LLS+19] Hyung Tae Lee, San Ling, Jae Hong Seo, and Huaxiong Wang. "Public key encryption with equality test from generic assumptions in the random oracle model." In: *Inf. Sci.* 500 (2019), pp. 15–33.

[LLS+20] Hyung Tae Lee, San Ling, Jae Hong Seo, Huaxiong Wang, and Taek-Young Youn. "Public key encryption with equality test in the standard model." In: *Inf. Sci.* 516 (2020), pp. 89–108.

[LMH+19] Yunhao Ling, Sha Ma, Qiong Huang, Ru Xiang, and Ximing Li. "Group ID-Based Encryption with Equality Test." In: *ACISP*. 2019, pp. 39–57.

[LOS+10] 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*. 2010, pp. 62–91.

[LSQ+21] Xi Jun Lin, Lin Sun, Haipeng Qu, and Xiaoshuai Zhang. "Public key encryption supporting equality test and flexible authorization without bilinear pairings." In: *Comput. Commun.* 170 (2021), pp. 190–199.

[LSQ18] Xi Jun Lin, Lin Sun, and Haipeng Qu. "Generic construction of public key encryption, identity-based encryption and signcryption with equality test." In: *Inf. Sci.* 453 (2018), pp. 111–126.

[LSX+21] Cong Li, Qingni Shen, Zhikang Xie, Xinyu Feng, Yuejian Fang, and Zhonghai Wu. "Large Universe CCA2 CP-ABE With Equality and Validity Test in the Standard Model." In: *Comput. J.* 64.4 (2021), pp. 509–533.

[LWS+21] Xi Jun Lin, Qihui Wang, Lin Sun, and Haipeng Qu. "Identity-based encryption with equality test and datestamp-based authorization mechanism." In: *Theor. Comput. Sci.* 861 (2021), pp. 117–132.

[LZL12] Yao Lu, Rui Zhang, and Dongdai Lin. "Stronger Security Model for Public-Key Encryption with Equality Test." In: *Pairing*. 2012, pp. 65–82.

[Ma16] Sha Ma. "Identity-based encryption with outsourced equality test in cloud computing." In: *Inf. Sci.* 328 (2016), pp. 389–402.

[MZH+15] Sha Ma, Mingwu Zhang, Qiong Huang, and Bo Yang. "Public Key Encryption with Delegated Equality Test in a Multi-User Setting." In: *Comput. J.* 58.4 (2015), pp. 986–1002.

[NSD+20] Giang Linh Duc Nguyen, Willy Susilo, Dung Hoang Duong, Huy Quoc Le, and Fuchun Guo. "Lattice-Based IBE with Equality Test Supporting Flexible Authorization in the Standard Model." In: *INDOCRYPT*. 2020, pp. 624–643.

[QYL+18] Haipeng Qu, Zhen Yan, Xi Jun Lin, Qi Zhang, and Lin Sun. "Certificateless public key encryption with equality test." In: *Inf. Sci.* 462 (2018), pp. 76–92.

[SDL20] Willy Susilo, Dung Hoang Duong, and Huy Quoc Le. "Efficient Post-quantum Identity-based Encryption with Equality Test." In: *ICPADS*. 2020, pp. 633–640.

[Tak21] Atsushi Takayasu. "Tag-based ABE in prime-order groups via pair encoding." In: *Des. Codes Cryptogr.* 89.8 (2021), pp. 1927–1963.

[Tan11] Qiang Tang. "Towards Public Key Encryption Scheme Supporting Equality Test with Fine-Grained Authorization." In: *ACISP*. 2011, pp. 389–406.

[TKN20] Junichi Tomida, Yuto Kawahara, and Ryo Nishimaki. "Fast, Compact, and Expressive Attribute-Based Encryption." In: *PKC*. 2020, pp. 3–33.

[WCH+20] Yuanhao Wang, Yuzhao Cui, Qiong Huang, Hongbo Li, Jianye Huang, and Guomin Yang. "Attribute-Based Equality Test Over Encrypted Data Without Random Oracles." In: *IEEE Access* 8 (2020), pp. 32891–32903.

[Wee14] Hoeteck Wee. "Dual System Encryption via Predicate Encodings." In: *TCC*. 2014, pp. 616–637.

[Xag13] Keita Xagawa. "Improved (Hierarchical) Inner-Product Encryption from Lattices." In: *PKC*. 2013, pp. 235–252.

[YAH+11] Shota Yamada, Nuttapong Attrapadung, Goichiro Hanaoka, and Noboru Kunihiro. "Generic Constructions for Chosen-Ciphertext Secure Attribute Based Encryption." In: *PKC*. 2011, pp. 71–89.

[YTH+10] Guomin Yang, Chik How Tan, Qiong Huang, and Duncan S. Wong. "Probabilistic Public Key Encryption with Equality Test." In: *CT-RSA*. 2010, pp. 119–131.

[ZCL+19] Kai Zhang, Jie Chen, Hyung Tae Lee, Haifeng Qian, and Huaxiong Wang. "Efficient public key encryption with equality test in the standard model." In: *Theor. Comput. Sci.* 755 (2019), pp. 65–80.

[ZCZ+19] Ming Zeng, Jie Chen, Kai Zhang, and Haifeng Qian. "Public key encryption with equality test via hash proof system." In: *Theor. Comput. Sci.* 795 (2019), pp. 20–35.