Conversions among Several Classes of Predicate Encryption and Their Applications

Shota Yamada[1], Nuttapong Attrapadung[1], and Goichiro Hanaoka[1]

National Institute of Advanced Industrial Science and Technology (AIST). {yamada-shota, n.attrapadung, hanaoka-goichiro }@aist.go.jp

Abstract. Predicate encryption is an advanced form of public-key encryption that yield high flexibility in terms of access control. In the literature, many predicate encryption schemes have been proposed such as fuzzy-IBE, KP-ABE, CP-ABE, (doubly) spatial encryption (DSE), and ABE for arithmetic span programs. In this paper, we study relations among them and show that some of them are in fact equivalent by giving conversions among them. More specifically, our main contributions are as follows:

– We show that monotonic, small universe KP-ABE (CP-ABE) with bounds on the size of attribute sets and span programs (or linear secret sharing matrix) can be converted into DSE. Furthermore, we show that DSE implies non-monotonic CP-ABE (and KP-ABE) with the same bounds on parameters. This implies that monotonic/non-monotonic KP/CP-ABE (with the bounds) and DSE are all equivalent in the sense that one implies another.
– We also show that if we start from KP-ABE without bounds on the size of span programs (but bounds on the size of attribute sets), we can obtain ABE for arithmetic span programs. The other direction is also shown: ABE for arithmetic span programs can be converted into KP-ABE. These results imply, somewhat surprisingly, KP-ABE without bounds on span program sizes is in fact equivalent to ABE for arithmetic span programs, which was thought to be more expressive or at least incomparable.

By applying these conversions to existing schemes, we obtain many non-trivial consequences. We obtain the first non-monotonic, large universe CP-ABE (that supports span programs) with constant-size ciphertexts, the first ABE for arithmetic span programs with adaptive security, the first ciphertext-policy version of ABE for arithmetic span programs, and even more. We also obtain the first attribute-based signature scheme that supports non-monotone span programs and achieves constant-size signatures via our technique.

Keywords. Attribute-based encryption (for arithmetic span programs), doubly spatial encryption, generic conversion, constant-size ciphertexts, constant-size keys

1 Introduction

Predicate encryption (PE) is an advanced form of public-key encryption that allows much flexibility. Instead of encrypting data to a target recipient, a sender will specify in a more general way about who should be able to view the message. In predicate encryption for a predicate R, a sender can associate a ciphertext with a ciphertext attribute Y

while a private key is associated with a key attribute X. Such a ciphertext can then be decrypted by such a key if the predicate evaluation $R(X, Y)$ holds true.

There exist many classes of PE, each is defined by specifying a corresponding class of predicates. One notable class is attribute-based encryption (ABE) [37,22] for span programs (or equivalently, linear secret sharing schemes), of which predicate is defined over key attributes being a span program and ciphertext attributes being a set of attributes, and its evaluation holds true if the span program accepts the set. This is called key-policy ABE (KP-ABE). There is also ciphertext-policy ABE (CP-ABE), where the roles of key and ciphertext attributes are exchanged. Another important class is doubly spatial encryption (DSE) [23], of which predicate is defined over both key and ciphertext attributes being affine subspaces, and its evaluation holds true if both subspaces intersect. Very recently, a new important class of PE, that is called attribute encryption for arithmetic span programs is defined in [26]. They showed such a PE scheme is useful by demonstrating that the scheme can be efficiently converted into ABE for arithmetic branching programs for both zero-type and non-zero type predicates. If the scheme satisfies a certain requirement for efficiency (namely, encryption cost is at most linear in ciphertext predicate size), it is also possible to obtain a publicly verifiable delegation scheme for arithmetic branching programs, by exploiting a conversion shown in [35]. Furthermore, they gave a concrete construction of such scheme.

Compared to specific constructions of predicate encryption schemes [28,29,31,41,19,17] (to name only a few) that aims at achieving more expressive predicates and/or stronger security guarantee, relations between predicate encryption schemes are much less investigated. The purpose of this paper is to improve our understanding of relations between them.

1.1 Our Results

Relations among PE. Towards the goal above, we study relations among PE and show that some of them are in fact equivalent by giving generic conversion among them. We first investigate the relation among ABE with some bounds on parameters (the size of attribute sets and the size of span programs) and DSE. We have the following results:

- At first, we show a conversion from KP-ABE (or CP-ABE) with the bounds on parameters into DSE (without key delegation, in Section 3). Such an implication is not straightforward in the first place. Intuitively, one reason stems from the different nature between both predicates: while DSE can be considered as an algebraic object that involves affine spaces, ABE can be seen as a somewhat more combinatorial object that involves sets (of attributes). Our approach involves some new technique for "programming" a set associated to a ciphertext and a span program associated to a private key in the KP-ABE scheme so that they can emulate the relation for doubly spatial encryption.
- We then extend the result of [23], which showed that DSE implies CP/KP-ABE with large universes. We provide a new conversion from DSE (without delegation) to non-monotonic CP/KP-ABE with large universes (in Section 4). We note that the resulting schemes obtained by the above conversions have some bounds on parameters. In the conversion, we extensively use a special form of polynomial introduced in [27] and carefully design a matrix so that DSE can capture a relation for ABE.

Somewhat surprisingly, by combining the above results, we obtain generic conversions that can boost the functionality of (bounded) ABE: from monotonic to non-monotonic, and from small-universe to large-universe; moreover, we also obtain conversions which transform ABE to its dual (key-policy to ciphertext-policy, and vice versa). This implies that they are essentially equivalent in some sense. See Figure 1 for the details.

So far, we have considered ABE schemes with bounds on parameters, especially on the size of span programs. Then we proceed to investigate relation among ABE schemes without bounds on the size of span programs (but with a bound on the size of attribute sets) and ABE for arithmetic span programs recently introduced and studied by Ishai and Wee [26]. We call the latter key-policy ABE for arithmetic span programs (KASP), since in the latter, a ciphertext is associated with a vector while a private key is associated with an arithmetic span program which specifies a policy. By exchanging key and ciphertext attribute, we can also define ciphertext-policy version of ABE for arithmetic span program (CASP). We have the following results:

– We show that monotonic KP-ABE with small universe (without bound on the size of span programs) can be converted into KASP (in Section 5). The idea for the conversion is similar to that in Section 3.
– Then, we proceed to investigate the converse direction. In fact, we can show somewhat stronger result. That is, we show that KASP can be converted into non-monotonic KP-ABE with large universe, which trivially implies monotonic KP-ABE with small universe (in Appendix E). The idea for the conversion is similar to that in Section 4.

Given the above results, we have all of the following are equivalent: monotonic KP-ABE with small universe, non-monotonic KP-ABE with large universe, and KASP. Similar implications hold for the case of CP-ABE and CASP. However, we do not have a conversion from KP-ABE to CP-ABE in this case. Again, see Figure 1 for the details.

Applications. By applying our conversions to existing schemes, we obtain many new schemes. Some of them have new properties that were not achieved before. See Section 6 and tables therein for the overview. We obtain following new DSE schemes by applying our KP(CP)-ABE-to-DSE conversion to existing schemes.

– From unbounded KP-ABE schemes [34,36], we obtain the first DSE with constant-size master public key (but without delegation). Furthermore, for the scheme obtained from [36], we can define a key delegation algorithm satisfying appropriate property.
– From KP-ABE schemes with constant-size ciphertexts [2,25,38], we can obtain the first DSE schemes with constant-size ciphertexts. We can define a key delegation algorithm for the scheme obtained from [2,25] too.

Furthermore, by applying the DSE-to-non-monotonic-CP/KP-ABE conversion to the DSE schemes obtained above, we also obtain several new schemes.

– From the DSE scheme with constant-size ciphertexts, we obtain the first non-monotonic and large universe CP-ABE scheme for span programs with constant-size ciphertexts. Note that previous CP-ABE schemes with constant-size ciphertexts [16,11,18,10] only support threshold or even more limited predicates. Our results significantly improve the expressibility.

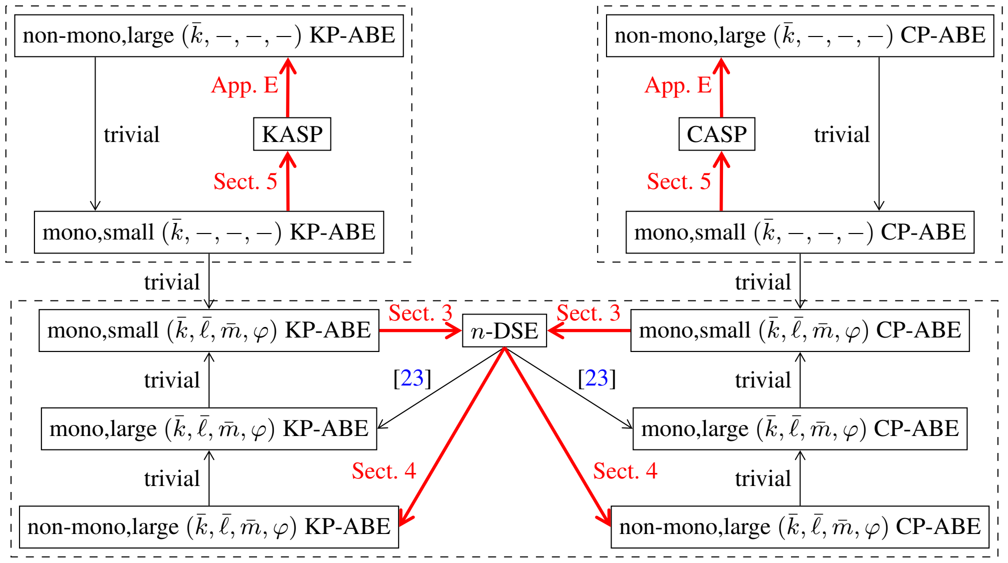
non-mono,large $(\bar{k},-,-,-)$ KP-ABE
App. E
KASP
trivial
Sect. 5
mono,small $(\bar{k},-,-,-)$ KP-ABE
non-mono,large $(\bar{k},-,-,-)$ CP-ABE
App. E
CASP
trivial
Sect. 5
mono,small $(\bar{k},-,-,-)$ CP-ABE
trivial
trivial
mono,small $(\bar{k},\bar{\ell},\bar{m},\varphi)$ KP-ABE
Sect. 3
n-DSE
Sect. 3
mono,small $(\bar{k},\bar{\ell},\bar{m},\varphi)$ CP-ABE
trivial
[23]
[23]
trivial
mono,large $(\bar{k},\bar{\ell},\bar{m},\varphi)$ KP-ABE
mono,large $(\bar{k},\bar{\ell},\bar{m},\varphi)$ CP-ABE
trivial
Sect. 4
Sect. 4
trivial
non-mono,large $(\bar{k},\bar{\ell},\bar{m},\varphi)$ KP-ABE
non-mono,large $(\bar{k},\bar{\ell},\bar{m},\varphi)$ CP-ABE

Fig. 1. Relations among predicate encryption primitives. In this figure, arrows indicate conversions that transform the primitive of the starting point to that of the end point. The red arrows indicate our results in this paper. For ABE, 'mono' and 'non-mono' indicates whether it is monotonic or non-monotonic, while 'small' and 'large' indicate whether the attribute universes are large (i.e., exponentially large) or small (i.e., polynomially bounded). $(\bar{k},\bar{\ell},\bar{m},\varphi)$ specify bounds on size of sets of attributes and span programs. See Section 2.1 for the details. As a result, primitives inside each dashed box are all equivalent in the sense there is a conversion between each pair of them.

Moreover, by applying the KP(CP)-ABE-to-K(C)ASP conversion to existing schemes, we also obtain new schemes. Note that the KASP scheme given in [26] is selectively secure and the size of master public keys and ciphertexts are linear in the dimension of the scheme.

- From unbounded the KP-ABE schemes in [36], we obtain the first KASP scheme with constant-size master public key.
- From the KP-ABE schemes with constant-size ciphertexts [2,38], we obtain the first KASP scheme with constant size ciphertexts.
- From the adaptively secure KP-ABE scheme [31], we obtain the first KASP scheme with adaptive security.
- From the CP-ABE schemes in [40,31], we obtain the first CASP schemes. In particular, the construction obtained from [31] achieve adaptive security.

Our technique is also useful in the settings of attribute-based signatures (ABS) [32,33]. We first define a notion that we call predicate signature (PS) which is a signature analogue of PE. Then, we construct a specific PS scheme with constant-size signatures such that a signature is associated with a set of attributes while a private key is associated with a policy (or monotone span programs). This is in some sense a dual notion of ordinary ABS in which a signature is associated with a policy and a private key with a set. By using the technique developed in the above, we can convert the PS scheme into an ABS scheme. As a result, we obtain the first ABS scheme with constant-size signatures. Previous ABS schemes with constant-size signatures [24,10] only support threshold or more limited classes of predicates.

Finally, we remark that although our conversions are feasible, they often introduce polynomial-size overheads to some parameters. Thus, in most cases, above schemes obtained by the conversions should be seen as feasibility results in the sense that they might not be totally efficient.

1.2 Related Works

There are several previous works investigating relations among PE primitives. In [21], a black box separation between threshold predicate encryption (fuzzy IBE) and IBE was shown. They also rule out certain natural constructions of PE for $\mathbf{NC}^1$ from PE for $\mathbf{AC}^0$. In [13], it was shown that hierarchical inner product encryption is equivalent to spatial encryption, which is a special case of doubly spatial encryption.

[20] showed a generic conversion from KP-ABE supporting threshold formulae to CP-ABE supporting threshold formulae. Their result and ours are incomparable. Our KP-ABE to CP-ABE conversion requires the original KP-ABE to support monotone span programs, which is a stronger requirement than [20]. On the other hand, the resulting scheme obtained by our conversion supports non-monotone span programs, which is a wider class than threshold formulae [1]. Thus, by applying our conversion, we can obtain new schemes (such as CP-ABE supporting non-monotone span programs with constant-size ciphertext) that is not possible to obtain by the conversion by [20].

In very recent works [1,3], it is shown that a PE scheme satisfying certain specific template can be converted into a PE scheme for its dual predicate. In particular, it yields KP-ABE-to-CP-ABE conversion. Again, their result and ours are incomparable. On the one hand, schemes obtained from their conversion are typically more efficient than ours. On the other hand, their conversion only works for schemes with the template while our conversion is completely generic. Furthermore, since they essentially exchange key and ciphertext components in the conversion, the size of keys and ciphertexts are also exchanged. For example, if we start from KP-ABE with constant-size ciphertexts, they obtain CP-ABE with *constant-size private keys* while we obtain CP-ABE with *constant-size ciphertexts*.

We also remark that in the settings where PE for general circuit is available, we can easily convert any KP-ABE into CP-ABE by using universal circuits as discussed in [19,17]. However, in the settings where only PE for span programs is available, this technique is not known to be applicable. We note that all existing PE schemes for general circuits [17,19,7] are quite inefficient and based on strong assumptions (e.g., existence of secure multi-linear map or hardness of certain lattice problems for an exponential approximation factor). In [5], in the context of quantum computation, Belovs studies a span program that decides whether two spaces intersect or not. The problem

[1] While it is known that monotone span programs contain threshold formulae [22], the converse is not known to be true.

and its solution considered there is very similar to that in Section 3 of our paper. However, he does not consider application to cryptography and the result is not applicable to our setting immediately since the syntax of span programs is slightly different.

2 Preliminaries

Notation. Throughout the paper, p denotes a prime number. We will treat a vector as a column vector, unless stated otherwise. For a vector $\mathbf{a} \in \mathbb{Z}_p^n$, $\mathbf{a}[i] \in \mathbb{Z}_p$ represents i-th element of the vector. Namely, $\mathbf{a} = (\mathbf{a}[1], \ldots, \mathbf{a}[n])^\top$. For $\mathbf{a}, \mathbf{b} \in \mathbb{Z}_p^n$, we denote their inner product as $\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^\top \mathbf{b} = \sum_{i=1}^n \mathbf{a}[i] \cdot \mathbf{b}[i]$. We denote by $\mathbf{e}_i$ the i-th unit vector: its i-th component is one, all others are zero. $\mathbf{I}_n$ and $\mathbf{0}_{n \times m}$ represent an identity matrix in $\mathbb{Z}_p^{n \times n}$ and zero matrix in $\mathbb{Z}_p^{n \times m}$ respectively. We also define $\mathbf{1}_n = (1, 1, \ldots, 1)^\top \in \mathbb{Z}_p^n$ and $\mathbf{0}_n = \mathbf{0}_{n \times 1}$. We often omit the subscript if it is clear from the context. We denote by $[a, b]$ a set $\{a, a+1, \ldots, b\}$ for $a, b \in \mathbb{Z}$ such that $a \leq b$ and $[b]$ denotes $[1, b]$. For a matrix $\mathbf{X} \in \mathbb{Z}_p^{n \times d}$, $\mathrm{span}(\mathbf{X})$ denotes a linear space $\{\mathbf{X} \cdot \mathbf{u} | \mathbf{u} \in \mathbb{Z}_p^d\}$ spanned by columns of $\mathbf{X}$. For matrices $\mathbf{A} \in \mathbb{Z}_p^{n_1 \times m}$ and $\mathbf{B} \in \mathbb{Z}_p^{n_2 \times m}$, $[\mathbf{A}; \mathbf{B}] \in \mathbb{Z}_p^{(n_1+n_2) \times m}$ denotes $[\mathbf{A}^\top, \mathbf{B}^\top]^\top$ i.e., vertical concatenation of them.

2.1 Definition of Predicate Encryption

Here, we define the syntax of predicate encryption. We emphasize that we do not consider attribute hiding in this paper[2].

Syntax. Let $R = \{R_N : A_N \times B_N \to \{0, 1\} \mid N \in \mathbb{N}^c\}$ be a relation family where A_N and B_N denote "ciphertext attribute" and "key attribute" spaces and c is some fixed constant. The index $N = (n_1, n_2, \ldots, n_c)$ of R_N denotes the numbers of bounds for corresponding parameters. A predicate encryption (PE) scheme for R is defined by the following algorithms:

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

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

$\mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, Y) \to \mathsf{sk}_Y$: The key generation algorithm takes as input the master secret key msk, the master public key mpk, and a key attribute $Y \in B_N$. It outputs a private key sk_Y.

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

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

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

Security. We now define the security for an PE scheme $\varPi$ by the following game between a challenger and an attacker $\mathcal{A}$.

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

A weaker notion called selective security can be defined as in the above game with the exception that the adversary $\mathcal{A}$ has to choose the challenge ciphertext index $X^\star$ before the setup phase but private key queries $Y_1, \ldots, Y_Q$ and choice of $(\mathsf{M}_0, \mathsf{M}_1)$ can still be adaptive.

2.2 (Arithmetic) Span Program, ABE, and Doubly Spatial Encryption

Definition 1 (Span Program). *Let $\mathcal{U} = \{u_1, \ldots, u_t\}$ be a set of variables. A span program over $\mathbb{Z}_p$ is a labelled matrix $(\mathbf{L}, \rho)$ where $\mathbf{L} \in \mathbb{Z}_p^{\ell \times m}$ and ρ is labelling of the rows of $\mathbf{L}$ by literals from $\{u_1, \ldots, u_t, \neg u_1, \ldots, \neg u_t\}$ (every row is labelled by one literal), i.e., $\rho : [1, \ell] \rightarrow \{u_1, \ldots, u_t, \neg u_1, \ldots, \neg u_t\}$. A span program accepts or rejects an input by the following criterion. For every input sequence $\delta \in \{0, 1\}^t$ define the sub-matrix $\mathbf{L}_\delta$ of $\mathbf{L}$ consisting of those rows whose labels are set to 1 by the input δ, i.e., either rows labelled by some u_i such that $\delta_i = 1$ or rows labelled by some $\neg u_i$ such that $\delta_i = 0$. The span program $(\mathbf{L}, \rho)$ accepts δ if and only if $\mathbf{e}_1 \in \mathrm{span}(\mathbf{L}_\delta^\top)$ (in other words, $\mathbf{e}_1^\top = (1, 0, \ldots, 0)$ is included in the space spanned by the rows of $\mathbf{L}_\delta$). A span program is called monotone if the labels of the rows are only the positive literals $\{u_1, \ldots, u_t\}$.*

Key-(Ciphertext) Policy Attribute-Based Encryption. Let $\mathcal{U}$ be universe of attributes. We define a relation R^{KP} on all span programs $(\mathbf{L}, \rho)$ over $\mathbb{Z}_p$ and all sets of attributes $S \subseteq \mathcal{U}$. We also define $\delta \in \{0, 1\}^t$ as an indicator vector corresponding to S. Namely, $\delta_i = 1$ if $u_i \in S$ and $\delta_i = 0$ otherwise. Then, $R^{\mathsf{KP}}(S, (\mathbf{L}, \rho)) = 1$ if the span program $(\mathbf{L}, \rho)$ accepts δ and $R^{\mathsf{KP}}(S, (\mathbf{L}, \rho)) = 0$ otherwise. Similarly, R^{CP} is defined as $R^{\mathsf{CP}}((\mathbf{L}, \rho), S) = 1$ iff $(\mathbf{L}, \rho)$ accepts δ.

Many existing KP-ABE scheme has some bounds on parameters. Let $N = (\bar{k}, \bar{\ell}, \bar{m}, \varphi)$ specify the corresponding bounds (the maximum numbers) on the size of attribute sets, the number of rows in $\mathbf{L}$, the number of columns in $\mathbf{L}$, and the number of allowed repetition of same value that appears in $\rho(1),\ldots,\rho(\ell)$ where $\ell(\leq \bar{\ell})$ is the number of rows in $\mathbf{L}$, respectively. When there is no restriction on corresponding parameter, we represent it by "$-$" such as $(k,-,-,-)$. We define A_N and B_N as every set of attributes and span program whose size is restricted by N, respectively. The relation R^{KP} can be restricted on $A_N \times B_N$ naturally. We call this relation R^{KP}_N. A predicate encryption for R^{KP}_N is called KP-ABE. By setting A'_N and B'_N as every span program and sets of attributes restricted by N and restricting R^{CP} on $A'_N \times B'_N$, we can define R^{CP}_N. A predicate encryption for R^{CP}_N is called CP-ABE.

We say the scheme supports small universe if $t = |\mathcal{U}|$ is polynomially bounded and large universe if t is exponentially large. We also say that the scheme is monotonic if span program is restricted to be monotone, and non-monotonic otherwise.

Attribute-Based Encryption for Arithmetic Span Programs [26]. $N = n$ is the dimension for the scheme. A_N is $A_N = \mathbb{Z}_p^n$. Let us define an arithmetic span program as a tuple $(\mathbf{Y},\mathbf{Z},\rho)$ where $\mathbf{Y},\mathbf{Z} \in \mathbb{Z}_p^{m\times\ell}$ and $\rho : [\ell] \to [n]$ is a map. There is no restriction on ℓ and m. B_N is a set of all arithmetic span programs. Then, R^{KASP}_N is defined as $R^{\mathsf{KASP}}_N(\mathbf{x},(\mathbf{Y},\mathbf{Z},\rho)) = 1$ iff $\mathbf{e}_1 \in \mathbb{Z}_p^m$ is in a space spanned by vectors $\{\mathbf{x}[\rho(j)]\cdot\mathbf{y}_j + \mathbf{z}_j\}_{j\in[\ell]}$ where $\mathbf{x}[\rho(j)]$ is the $\rho(j)$-th coefficient of $\mathbf{x}$, and $\mathbf{y}_j$ and $\mathbf{z}_j$ are the j-th column of $\mathbf{Y}$ and $\mathbf{Z}$ respectively. We call predicate encryption for R^{KASP} key-policy attribute-based encryption for arithmetic span program (KASP). As in the case of KP-ABE, we can define its dual. Namely, we set A'_N as a set of all arithmetic span programs, B'_N as $\mathbb{Z}_p^n$, and define $R^{\mathsf{CASP}}_N : A'_N \times B'_N \to \{0,1\}$ as $R^{\mathsf{CASP}}_N(\mathbf{x},(\mathbf{Y},\mathbf{Z},\rho)) = 1$ iff $R^{\mathsf{KASP}}_N((\mathbf{Y},\mathbf{Z},\rho),\mathbf{x}) = 1$. We call predicate encryption for R^{CASP} ciphertext-policy attribute-based encryption for arithmetic span programs (CASP).

Doubly Spatial Encryption. $N = n$ is the dimension of the scheme. We define $A_N = B_N = \mathbb{Z}_p^n \times (\cup_{0\leq d\leq n}\mathbb{Z}_p^{n\times d})$. R^{DSE}_N is defined as $R^{\mathsf{DSE}}_N\big((\mathbf{x}_0,\mathbf{X}),(\mathbf{y}_0,\mathbf{Y})\big) = 1$ iff $\big(\mathbf{x}_0 + \mathrm{span}(\mathbf{X})\big) \cap \big(\mathbf{y}_0 + \mathrm{span}(\mathbf{Y})\big) \neq \emptyset$. Doubly spatial encryption is PE for relation R^{DSE}_N equipped with additional key delegation algorithm defined below.

$\mathsf{KeyDel}(\mathsf{mpk},\mathsf{sk}_{(\mathbf{y}_0,\mathbf{Y})},(\mathbf{y}_0,\mathbf{Y}),(\hat{\mathbf{y}}_0,\hat{\mathbf{Y}})) \to \mathsf{sk}_{\hat{\mathbf{Y}}}$: The key delegation algorithm takes as input the public parameters mpk, a private key $\mathsf{sk}_{(\mathbf{y}_0,\mathbf{Y})}$, $(\mathbf{y}_0,\mathbf{Y})$, and $(\hat{\mathbf{y}}_0,\hat{\mathbf{Y}})$ such that $\hat{\mathbf{y}}_0 + \mathrm{span}(\hat{\mathbf{Y}}) \subseteq \mathbf{y}_0 + \mathrm{span}(\mathbf{Y})$. It outputs a private key $\mathsf{sk}_{(\hat{\mathbf{y}}_0,\hat{\mathbf{Y}})}$.

We require that delegation is independent of the path taken. That is, for all λ, N, $(\mathsf{mpk},\mathsf{msk}) \leftarrow \mathsf{Setup}(\lambda,N)$, $\mathbf{y}_0,\mathbf{Y},\hat{\mathbf{y}}_0,\hat{\mathbf{Y}}$ such that $\hat{\mathbf{y}}_0 + \mathrm{span}(\hat{\mathbf{Y}}) \subseteq \mathbf{y}_0 + \mathrm{span}(\mathbf{Y})$, and $\mathsf{sk}_{(\mathbf{y}_0,\mathbf{Y})} \leftarrow \mathsf{KeyGen}(\mathsf{msk},\mathsf{mpk},(\mathbf{y}_0,\mathbf{Y}))$, we have the following distributions are identical: $\mathsf{KeyDel}(\mathsf{mpk},\mathsf{sk}_{(\mathbf{y}_0,\mathbf{Y})},(\mathbf{y}_0,\mathbf{Y}),(\hat{\mathbf{y}}_0,\hat{\mathbf{Y}})) \approx \mathsf{KeyGen}(\mathsf{mpk},\mathsf{msk},(\hat{\mathbf{y}}_0,\hat{\mathbf{Y}}))$.

2.3 Embedding Lemma for PE

The following useful lemma from [8] describes a sufficient criterion for implication for PE. The lemma is applicable to any relation family. We prove the lemma in Appendix A for completeness.

Lemma 1 (Embedding lemma [8]). *Consider two relation families* $R^{\mathsf{F}}_N : A_N \times B_N \to \{0,1\}$ *and* $R^{\mathsf{F}'}_{N'} : A'_{N'} \times B'_{N'} \to \{0,1\}$, *which is parametrized by* $N \in \mathbb{N}^c$ *and*

$N' \in \mathbb{N}^{c'}$ respectively. Suppose that there exist efficient mappings $f_{\mathsf{p}} : \mathbb{Z}^{c'} \to \mathbb{Z}^{c}$, $f_{\mathsf{e}} : A'_{N'} \to A_{f_{\mathsf{p}}(N')}$, and $f_{\mathsf{k}} : B'_{N'} \to B_{f_{\mathsf{p}}(N')}$ such that

$$R^{\mathsf{F}'}_{N'}(X', Y') = 1 \Leftrightarrow R^{\mathsf{F}}_{f_{\mathsf{p}}(N')}(f_{\mathsf{e}}(X'), f_{\mathsf{k}}(Y')) = 1.$$

Then, a PE $\Pi = \{\mathsf{Setup}, \mathsf{Encrypt}, \mathsf{KeyGen}, \mathsf{Decrypt}\}$ for R^{F}_{N} can be converted into a PE $\Pi' = \{\mathsf{Setup}', \mathsf{Encrypt}', \mathsf{KeyGen}', \mathsf{Decrypt}'\}$ for $R^{\mathsf{F}'}_{N'}$ as $\mathsf{Setup}'(\lambda, N') = \mathsf{Setup}(\lambda, f_{\mathsf{p}}(N'))$, $\mathsf{Encrypt}'(\mathsf{mpk}, \mathsf{M}, X') = \mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}, f_{\mathsf{e}}(X'))$, $\mathsf{KeyGen}'(\mathsf{msk}, \mathsf{mpk}, Y') = \mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, f_{\mathsf{k}}(Y'))$, and $\mathsf{Decrypt}'(\mathsf{mpk}, C, X', \mathsf{sk}_{Y'}, Y') = \mathsf{Decrypt}(\mathsf{mpk}, C, f_{\mathsf{e}}(X'), \mathsf{sk}_{Y'}, f_{\mathsf{k}}(Y'))$. Then, if Π is secure, so is Π'. This holds for selective security and adaptive security.

3 Conversion from ABE to DSE

In this section, we show how to construct DSE from monotonic KP-ABE (with bounds on the size of attribute sets and span programs). We note that by simply swapping key and ciphertext attributes, we can also obtain CP-ABE-to-DSE conversion. In the following, We first describe the conversion, then explain the intuition behind the conversion.

3.1 The Conversion

We construct DSE for dimension n without delegation from monotonic KP-ABE for parameters $N = (\bar{k}, \bar{\ell}, \bar{m}, \psi) = (n(n+1)\kappa + 1, 2(n\kappa + 1)(n+1), (n\kappa + 1)(n+1) + 1, 2(n+1))$ and the size of attribute universe is $|\mathcal{U}| = 2n(n+1)\kappa + 1$. Here, $\kappa = \lceil \log_2 p \rceil$. According to Lemma 1, it suffices to define appropriate maps $f^{\mathsf{DSE}\to\mathsf{KP}}_{\mathsf{p}}$, $f^{\mathsf{DSE}\to\mathsf{KP}}_{\mathsf{e}}$, and $f^{\mathsf{DSE}\to\mathsf{KP}}_{\mathsf{k}}$. We define $f^{\mathsf{DSE}\to\mathsf{KP}}_{\mathsf{p}}(n) = N$. $f^{\mathsf{DSE}\to\mathsf{KP}}_{\mathsf{e}}$ maps ciphertext attribute of DSE to that of KP-ABE while $f^{\mathsf{DSE}\to\mathsf{KP}}_{\mathsf{k}}$ maps key attribute of DSE to that of KP-ABE.

We set universe of attributes as $\mathcal{U} = \big\{\mathsf{Att}[i][j][k][b] \mid (i,j,k,b) \in [0,n] \times [1,n] \times [1,\kappa] \times \{0,1\}\big\} \cup \{\mathsf{D}\}$. Intuitively, $\mathsf{Att}[i][j][k][b]$ represents "k-th least significant bit of binary representation of j-th element of a vector $\mathbf{x}_i$ is $b \in \{0,1\}$". D is a dummy attribute which will be assigned for all ciphertext.

For $\mathbf{x}_0 \in \mathbb{Z}_p^n$ and $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_{d_1}] \in \mathbb{Z}_p^{n \times d_1}$ such that $d_1 \le n$, we define a set $S = f^{\mathsf{DSE}\to\mathsf{KP}}_{\mathsf{e}}(\mathbf{x}_0, \mathbf{X})$ as

$$S = \left\{\mathsf{Att}[i][j][k][b] \mid (i,j,k) \in [0,d_1] \times [1,n] \times [1,\kappa],\ b = \mathbf{x}_i[j][k]\right\} \cup \{\mathsf{D}\}.$$

In the above, we define $\mathbf{x}_i[j][k] \in \{0,1\}$ so that they satisfy $\mathbf{x}_i[j] = \sum_{k=1}^{\kappa} 2^{k-1} \cdot \mathbf{x}_i[j][k]$. Namely, $\mathbf{x}_i[j][k]$ is the k-th least significant bit of the binary representation of $\mathbf{x}_i[j] \in \mathbb{Z}_p$.

For $\mathbf{y}_0 \in \mathbb{Z}_p^n$ and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_{d_2}] \in \mathbb{Z}_p^{n \times d_2}$ such that $d_2 \leq n$, we define $f_{\mathsf{k}}^{\mathsf{DSE} \to \mathsf{KP}}(\mathbf{y}_0, \mathbf{Y}) = (\mathbf{L}, \rho)$ as follows. At first, $\mathbf{L}$ is defined as

$$\mathbf{L} = \begin{pmatrix} \mathbf{e}_1 & \mathbf{e}_1 + \mathbf{e}_{d_2+2} & \mathbf{y}_0^\top & & & & \\ & & \mathbf{Y}^\top & & & & \\ & & \mathbf{E} & \mathbf{J} & & & \\ & & \mathbf{E} & & \mathbf{J} & & \\ & & \vdots & & & \ddots & \\ & & \mathbf{E} & & & & \mathbf{J} \end{pmatrix} \in \mathbb{Z}_p^{\big((2n\kappa+1)(n+1)+d_2+1\big) \times \big((n\kappa+1)(n+1)+1\big)}. \tag{1}$$

In the above,

$$\mathbf{E} = \begin{pmatrix} \mathbf{g} & & & \\ & \mathbf{g} & & \\ & & \ddots & \\ & & & \mathbf{g} \\ 0 & 0 & \ldots & 0 \end{pmatrix} \in \mathbb{Z}_p^{(2n\kappa+1) \times n}, \ \mathbf{J} = \begin{pmatrix} -1 & & & \\ -1 & & & \\ & -1 & & \\ & -1 & & \\ & & \ddots & \\ & & & -1 \\ & & & -1 \\ 1 & 1 & \ldots & 1 \end{pmatrix} \in \mathbb{Z}_p^{(2n\kappa+1) \times n\kappa} \tag{2}$$

where $\mathbf{g} = (0, 1, 0, 2, \ldots, 0, 2^i, \ldots, 0, 2^{\kappa-1})^\top \in \mathbb{Z}_p^{2\kappa}$. In the matrix $\mathbf{L}$, each of $\mathbf{E}$ and $\mathbf{J}$ appears $n+1$ times. The leftmost and the second leftmost column of the matrix are $\mathbf{e}_1$ and $\mathbf{e}_1 + \mathbf{e}_{d_2+2}$ respectively. We also define a map $\rho : [1, \big(n'(n+1) + d_2 + 1\big)] \to \mathcal{U}$ which associates each row of $\mathbf{L}$ with an attribute. Here, we set $n' = 2n\kappa + 1$. At first, if $i \leq d_2 + 1$, $\rho(i) = \mathsf{D}$. Otherwise, $i \in [d_2 + 2, n'(n+1) + d_2 + 1]$ is represented as $i = (d_2 + 1) + n'i' + i''$ using unique $i' \in [0, n+1]$ and $i'' \in [0, n' - 1]$. If $i'' = 0$, $\rho(i)$ is set as $\rho(i) = \mathsf{D}$. Otherwise, $i'' \in [1, n' - 1 = 2n\kappa]$ can be represented as $i'' - 1 = 2\kappa j' + 2k' + b'$ using unique $j' \in [0, n-1]$, $k' \in [0, \kappa - 1]$, and $b' \in \{0, 1\}$. Then $\rho(i)$ is defined as $\rho(i) = \mathsf{Att}[i'][j'+1][k'+1][b']$.

Intuition. We explain the intuition behind the conversion. S can be seen as a binary representation of the information of $(\mathbf{x}_0, \mathbf{X})$. In the span program $(\mathbf{L}, \rho)$, $\mathbf{E}$ is used to reproduce the information of $(\mathbf{x}_0, \mathbf{X})$ in the matrix while $\mathbf{J}$ is used to constrain the form of linear combination among rows to a certain form. Note that a somewhat similar technique to ours that restricts the form of linear combination of vectors was used in [6] in a different context (for constructing a monotone span program that tests co-primality of two numbers). In some sense, the roll of the lower part of the matrix $\mathbf{L}$ (the last $n'(n+1)$ rows) is similar to universal circuit while the upper part of the matrix contains the information of $(\mathbf{y}_0, \mathbf{Y})$.

3.2 Correctness of the Conversion

By combining the following theorem and Lemma 1, we can see that KP-ABE for $N = (\bar{k}, \bar{\ell}, \bar{m}, \psi) = (n(n+1)\kappa + 1, 2(n\kappa+1)(n+1), (n\kappa+1)(n+1) + 1, 2(n+1))$ can be converted into DSE for dimension n (without delegation). The proof will appear in Appendix B.

Theorem 1. *For any $\mathbf{x}_0 \in \mathbb{Z}_p^n$, $\mathbf{X} \in \mathbb{Z}_p^{n\times d_1}$, $\mathbf{y}_0 \in \mathbb{Z}_p^n$ and $\mathbf{Y} \in \mathbb{Z}_p^{n\times d_2}$, it holds that*

$$R^{\mathsf{KP}}(S, (\mathbf{L}, \rho)) = 1 \Leftrightarrow R^{\mathsf{DSE}}\big((\mathbf{x}_0, \mathbf{X}), (\mathbf{y}_0, \mathbf{Y})\big) = 1$$

where $S = f_{\mathsf{e}}^{\mathsf{DSE}\to\mathsf{KP}}(\mathbf{x}_0, \mathbf{X})$ and $(\mathbf{L}, \rho) = f_{\mathsf{k}}^{\mathsf{DSE}\to\mathsf{KP}}(\mathbf{y}_0, \mathbf{Y})$ in the above.

4 From DSE to Non-Monotonic ABE

In [23], it is shown that DSE can be converted into monotonic CP-ABE with large universe (and bounds on the size of attribute sets and span programs). In this section, we extend their result to show that non-monotonic CP-ABE with large universe and the same bounds can be constructed from DSE. We note that our transformation is very diffenrent from that of [23] even if we only consider monotonic CP-ABE because of expositional reasons. We also note that by simply swapping key and ciphertext attributes, we immediately obtain DSE-to-non-monotonic-KP-ABE conversion.

4.1 The Conversion

We construct non-monotonic CP-ABE with large universe for $N = (\bar{k}, \bar{\ell}, \bar{m}, \bar{\ell})$ from DSE with dimension $n = 4\bar{\ell} + \bar{m} + 2\bar{k}\bar{\ell}$. As in Section 3, we define appropriate maps $f_{\mathsf{p}}^{\mathsf{CP}\to\mathsf{DSE}}$, $f_{\mathsf{e}}^{\mathsf{CP}\to\mathsf{DSE}}$, $f_{\mathsf{k}}^{\mathsf{CP}\to\mathsf{DSE}}$. At first, we define $f_{\mathsf{p}}^{\mathsf{CP}\to\mathsf{DSE}}(N) = n$. We assume that the universe of attributes is $\mathbb{Z}_p$. This restriction can be easily removed by using collision resistant hash. We first describe the conversion, then we explain the intuition.

Next we define $f_{\mathsf{e}}^{\mathsf{CP}\to\mathsf{DSE}}$ that takes as input a span program $(\mathbf{L}, \rho)$ and outputs pair of a vector and a matrix $(\mathbf{x}_0, \mathbf{X})$. Let the size of $\mathbf{L}$ be $(\ell_0+\ell_1)\times\bar{m}$ where $\ell_0+\ell_1 \le \bar{\ell}$. (If the number of columns is smaller, we can adjust the size by padding zeroes.) Without loss of generality, we assume that the first ℓ_0 rows of the matrix are associated with positive attributes and the last ℓ_1 rows with negative attributes by a map ρ. We denote $\mathbf{L}$ as $\mathbf{L} = [\mathbf{L}_0; \mathbf{L}_1]$ using matrices $\mathbf{L}_0 \in \mathbb{Z}_p^{\ell_0\times\bar{m}}$ and $\mathbf{L}_1 \in \mathbb{Z}_p^{\ell_1\times\bar{m}}$. $f_{\mathsf{e}}^{\mathsf{CP}\to\mathsf{DSE}}(\mathbf{L}, \rho) = (\mathbf{x}_0, \mathbf{X})$ is defined as follows:

$$\mathbf{x}_0 = -\mathbf{e}_1 \in \mathbb{Z}_p^n, \quad \mathbf{X}^\top = \begin{pmatrix} \mathbf{L}_0 & \overbrace{\hphantom{000}}^{\bar{\ell}} & \mathbf{G}_0 & & & \\ \mathbf{L}_1 & & & \mathbf{I}_{\ell_1} & \overbrace{\hphantom{000}}^{\bar{\ell}-\ell_1} & \\ & & & & & \mathbf{G}_1 \end{pmatrix} \in \mathbb{Z}_p^{(\ell_0+2\ell_1)\times n}.$$

In the above, $\mathbf{G}_b \in \mathbb{Z}_p^{\ell_b\times\bar{\ell}(\bar{k}+1)}$ for $b \in \{0,1\}$ are defined as

$$\mathbf{G}_b = \begin{pmatrix} \mathbf{p}\big(\rho(b\ell_0+1)\big)^\top & & & & \overbrace{\hphantom{000}}^{(\bar{\ell}-\ell_b)(\bar{k}+1)} \\ & \mathbf{p}\big(\rho(b\ell_0+2)\big)^\top & & & \\ & & \ddots & & \\ & & & \mathbf{p}\big(\rho(b\ell_0+\ell_b)\big)^\top & \end{pmatrix} \tag{3}$$

where $\mathbf{p}()$ is a function that takes an element of $\mathbb{Z}_p$ or its negation ($\{\neg x | x \in \mathbb{Z}_p\}$) as an input and outputs a vector $\mathbf{p}(x) = (1, x, x^2, \ldots, x^{\bar{k}})^\top \in \mathbb{Z}_p^{\bar{k}+1}$.

Next we define $f_{\mathsf{k}}^{\mathsf{CP}\to\mathsf{DSE}}$ that takes as input a set $S = (S_1, \ldots, S_k)$ such that $k \leq \bar{k}$ and outputs a vector $\mathbf{y}_0$ and a matrix $\mathbf{Y}$. $f_{\mathsf{k}}^{\mathsf{CP}\to\mathsf{DSE}}(S) = (\mathbf{y}_0, \mathbf{Y})$ is defined as follows.

$$\mathbf{y}_0 = \mathbf{0}_n \in \mathbb{Z}_p^n, \qquad \mathbf{Y}^\top = \begin{pmatrix} \overbrace{\phantom{\mathbf{0}}}^{\bar{m}} & \mathbf{H} & \mathbf{I}_{(\bar{k}+1)\bar{\ell}} & \\ & & & \mathbf{H} & \mathbf{I}_{(\bar{k}+1)\bar{\ell}} \end{pmatrix} \in \mathbb{Z}_p^{2(\bar{k}+1)\bar{\ell}\times n}.$$

In the above, $\mathbf{H}$ is defined as

$$\mathbf{H} = \mathbf{I}_{\bar{\ell}} \otimes \mathbf{q}_S = \begin{pmatrix} \mathbf{q}_S & & & \\ & \mathbf{q}_S & & \\ & & \ddots & \\ & & & \mathbf{q}_S \end{pmatrix} \in \mathbb{Z}_p^{\left((\bar{k}+1)\bar{\ell}\right)\times\bar{\ell}}$$

where $\mathbf{q}_S = (\mathbf{q}_S[1], \ldots, \mathbf{q}_S[\bar{k}+1])^\top \in \mathbb{Z}_p^{\bar{k}+1}$ is defined as a coefficient vector from $Q_S[Z] = \sum_{i=1}^{k+1} \mathbf{q}_S[i] \cdot Z^{i-1} = \prod_{i=1}^{k}(Z - S_i)$. If $k < \bar{k}$, the coordinates $\mathbf{q}_S[k+2], \ldots, \mathbf{q}_S[\bar{k}+1]$ are all set to 0.

Intuition. The matrices $\mathbf{X}$ and $\mathbf{Y}$ constructed above can be divided into two parts. The first ℓ_0 rows of $\mathbf{X}^\top$ and the first $(\bar{k}+1)\bar{\ell}$ rows of $\mathbf{Y}^\top$ deal with positive attributes.The lower parts of $\mathbf{X}^\top$ and $\mathbf{Y}^\top$ deal with negation of attributes. Here, we explain how we handle negated attributes. Positive attributes are handled by a similar mechanism. $\mathbf{I}_{(\bar{k}+1)\bar{\ell}}$ in $\mathbf{Y}^\top$ and $\mathbf{G}_1$ in $\mathbf{X}^\top$ restricts the linear combination of the rows of $\mathbf{X}^\top$ and $\mathbf{Y}^\top$ to a certain form in order to two affine spaces to have a intersection. As a result, we can argue that the coefficient of the i-th row of $\mathbf{L}_1$ in the linear combination should be multiple of $Q_S(\rho(\ell_0 + i))$[3]. Since we have that $Q_S(x) = 0$ iff $x \in S$ for any $x \in \mathbb{Z}_p$, this means that the coefficient of the vector in the linear combination should be 0 if $\rho(\ell_0 + i) = \neg\mathsf{Att}$ and $\mathsf{Att} \in S$. This restriction is exactly what we need to emulate predicate of non-monotonic CP-ABE.

4.2 Correctness of the Conversion

By combining the following theorem with Lemma 1, we can see that non-monotonic CP-ABE with large universe can be constructed from DSE.

Theorem 2. *For any span program $(\mathbf{L} \in \mathbb{Z}_p^{\ell\times m}, \rho)$ such that $\ell \leq \bar{\ell}$ and $m \leq \bar{m}$ and S such that $|S| \leq \bar{k}$, we have that*

$$R^{\mathsf{DSE}}((\mathbf{x}_0, \mathbf{X}), (\mathbf{y}_0, \mathbf{Y})) = 1 \Leftrightarrow R^{\mathsf{CP}}(S, (\mathbf{L}, \rho)) = 1$$

where $(\mathbf{x}_0, \mathbf{X}) = f_{\mathsf{e}}^{\mathsf{CP}\to\mathsf{DSE}}(\mathbf{L}, \rho)$ and $(\mathbf{y}_0, \mathbf{Y}) = f_{\mathsf{k}}^{\mathsf{CP}\to\mathsf{DSE}}(S)$ is defined as above.

[3] Here, We treat negated attributes ($\{\neg x | x \in \mathbb{Z}_p\}$) as elements of $\mathbb{Z}_p$. Namely, if $\rho(\ell_0 + i) = \neg\mathsf{Att}$ for some $\mathsf{Att} \in \mathbb{Z}_p$, $Q_S(\rho(\ell_0 + i)) := Q_S(\mathsf{Att})$.

Proof. Let $I \subset [1, \ell_0 + \ell_1]$ be $I = \{i | (\rho(i) = \mathsf{Att} \wedge \mathsf{Att} \in S) \vee (\rho(i) = \neg\mathsf{Att} \wedge \mathsf{Att} \notin S)\}$. We also let $\mathbf{L}_I$ be the sub-matrix of $\mathbf{L}$ formed by rows whose index is in I. Then, it suffices to show that

Claim. $\big(\mathbf{x}_0 + \mathrm{span}(\mathbf{X})\big) \cap \big(\mathbf{y}_0 + \mathrm{span}(\mathbf{Y})\big) \neq \emptyset \Leftrightarrow \mathbf{e}_1 \in \mathrm{span}(\mathbf{L}_I^\top)$.

Proof (Proof of the Claim). We show both directions separately in the following.
The Forward Direction ($\Rightarrow$). Let us assume that there exist $\mathbf{u} \in \mathbb{Z}_p^{\ell_0+2\ell_1}$ and $\mathbf{v} \in \mathbb{Z}_p^{2(\bar{k}+1)\bar{\ell}}$ such that $\mathbf{x}_0^\top + \mathbf{u}^\top\mathbf{X}^\top = \mathbf{y}_0^\top + \mathbf{v}^\top\mathbf{Y}^\top = \mathbf{v}^\top\mathbf{Y}^\top$. We denote these vectors as

$$\mathbf{u}^\top = (\underbrace{\mathbf{u}_0^\top}_{\ell_0}, \underbrace{\mathbf{u}_1^\top}_{\ell_1}, \underbrace{\mathbf{u}_2^\top}_{\ell_1}) \qquad \mathbf{v}^\top = (\underbrace{\mathbf{v}_1^\top}_{\bar{k}+1}, \dots, \underbrace{\mathbf{v}_{\bar{\ell}}^\top}_{\bar{k}+1}\, \underbrace{\mathbf{w}_1^\top}_{\bar{k}+1}, \dots, \underbrace{\mathbf{w}_{\bar{\ell}}^\top}_{\bar{k}+1}.)$$

Then, $\mathbf{x}_0^\top + \mathbf{u}^\top\mathbf{X}$ and $\mathbf{v}^\top\mathbf{Y}$ are written as

$$\begin{aligned}\mathbf{x}_0^\top + \mathbf{u}^\top\mathbf{X} = \Big(&\underbrace{-\mathbf{e}_1^\top + \mathbf{u}_0^\top\mathbf{L}_0 + \mathbf{u}_1^\top\mathbf{L}_1}_{\bar{m}}, \mathbf{0}_{\bar{\ell}}^\top, \underbrace{\mathbf{u}_0[1]\cdot\mathbf{p}(\rho(1))^\top, \dots, \mathbf{u}_0[\ell_0]\cdot\mathbf{p}(\rho(\ell_0))^\top}_{(\bar{k}+1)\ell_0}, \mathbf{0}_{(\bar{\ell}-\ell_0)(\bar{k}+1)}^\top,\\ &\underbrace{\mathbf{u}_1^\top}_{\ell_1}, \mathbf{0}_{\bar{\ell}-\ell_1}^\top, \underbrace{\mathbf{u}_2[1]\cdot\mathbf{p}(\rho(\ell_0+1))^\top, \dots, \mathbf{u}_2[\ell_1]\cdot\mathbf{p}(\rho(\ell_0+\ell_1))^\top}_{(\bar{k}+1)\ell_1}, \mathbf{0}_{(\bar{\ell}-\ell_1)(\bar{k}+1)}^\top\Big) \end{aligned} \tag{4}$$

and

$$\begin{aligned}\mathbf{v}^\top\mathbf{Y} = (\mathbf{0}_{\bar{m}}^\top, &\underbrace{\langle\mathbf{v}_1, \mathbf{q}_S\rangle, \dots,, \langle\mathbf{v}_{\bar{\ell}}, \mathbf{q}_S\rangle}_{\bar{\ell}}, \underbrace{\mathbf{v}_1^\top, \dots, \mathbf{v}_{\bar{\ell}}^\top}_{(\bar{k}+1)\bar{\ell}},\\ &\underbrace{\langle\mathbf{w}_1, \mathbf{q}_S\rangle, \dots,, \langle\mathbf{w}_{\bar{\ell}}, \mathbf{q}_S\rangle}_{\bar{\ell}}, \underbrace{\mathbf{w}_1^\top, \dots, \mathbf{w}_{\bar{\ell}}^\top}_{(\bar{k}+1)\bar{\ell}}).\end{aligned} \tag{5}$$

At first, by comparing the $\bar{m} + \bar{\ell} + 1$-th to $\bar{m} + (\bar{k}+2)\bar{\ell}$-th elements of the vector, we obtain that $\mathbf{v}_i = \mathbf{u}_0[i]\cdot\mathbf{p}(\rho(i))$ for $i \in [1, \ell_0]$ and $\mathbf{v}_i = \mathbf{0}_{\bar{k}+1}$ for $i \in [\ell_0+1, \bar{\ell}]$. Furthermore, by comparing $\bar{m}+1$-th to $\bar{m}+\bar{\ell}$-th elements of the vector, we have

$$\langle\mathbf{v}_i, \mathbf{q}_S\rangle = \mathbf{u}_0[i]\cdot\langle\mathbf{p}(\rho(i)), \mathbf{q}_S\rangle = \mathbf{u}_0[i]\cdot Q_S\big(\rho(i)\big) = 0$$

for $i \in [1, \ell_0]$. The second equation above follows from the definition of $\mathbf{p}()$ and $\mathbf{q}_S$. Since $Q_S(\rho(i)) = \prod_{\omega\in S}(\rho(i)-\omega) \neq 0$ if $\rho(i) \notin S$, we have that $\mathbf{u}_0[i] = 0$ if $\rho(i) \notin S$. That is, $\mathbf{u}_0[i] = 0$ for $i \in [1, \ell_0]\backslash I$.

Next, by comparing the last $(\bar{k}+1)\bar{\ell}$ elements in the vector, we obtain that $\mathbf{w}_i = \mathbf{u}_2[i]\cdot\mathbf{p}(\rho(\ell_0+i))$ for $i \in [1, \ell_1]$ and $\mathbf{w}_i = \mathbf{0}_{\bar{k}+1}$ for $i \in [\ell_1+1, \bar{\ell}]$. By comparing the $\bar{m} + (\bar{k}+2)\bar{\ell} + 1$-th to $\bar{m} + (\bar{k}+3)\bar{\ell}$-th elements in the vector, we have that $(\mathbf{u}_1^\top, \mathbf{0}_{\bar{\ell}-\ell_1}^\top) = (\langle\mathbf{w}_1, \mathbf{q}_S\rangle, \dots,, \langle\mathbf{w}_{\bar{\ell}}, \mathbf{q}_S\rangle)$ and thus

$$\mathbf{u}_1[i] = \langle\mathbf{w}_i, \mathbf{q}_S\rangle = \mathbf{u}_2[i]\cdot\langle\mathbf{p}(\rho(\ell_0+i)), \mathbf{q}_S\rangle = \mathbf{u}_2[i]\cdot Q_S(\rho(\ell_0+i))$$

holds for $i \in [1, \ell_1]$. From the above, we have that $\mathbf{u}_1[i] = 0$ if $\rho(\ell_0+i) = \neg\mathsf{Att}$ and $\mathsf{Att} \in S$ for some Att. This implies that $\mathbf{u}_1[i] = 0$ if $(\ell_0+i) \notin I$ for $i \in [1, \ell_1]$.

Finally, by comparing the first $\bar{m}$ elements in the vector, we obtain that $-\mathbf{e}_1^\top + \mathbf{u}_0^\top \mathbf{L}_0 + \mathbf{u}_1^\top \mathbf{L}_1 = \mathbf{0}^\top$. Let $\mathbf{u}_{0,I}$ be a subvector of $\mathbf{u}_0$ which is obtained by deleting all elements $\mathbf{u}_0[i]$ for $i \notin I$. Similarly, we define $\mathbf{u}_{1,I}$ as a vector obtained by deleting all elements $\mathbf{u}_1[i]$ for i such that $(\ell_0 + i) \notin I$ from $\mathbf{u}_1$. Since $\mathbf{u}_0[i] = 0$ for $i \in [1, \ell_0] \backslash I$ and $\mathbf{u}_1[i] = 0$ for $i \in [1, \ell_1]$ such that $(\ell_0 + i) \notin I$, it follows that $(\mathbf{u}_{0,I}^\top, \mathbf{u}_{1,I}^\top)\mathbf{L}_I = \mathbf{u}_0^\top \mathbf{L}_0 + \mathbf{u}_1^\top \mathbf{L}_1 = \mathbf{e}_1^\top$ and thus $\mathbf{e}_1 \in \mathrm{span}(\mathbf{L}_I^\top)$ as desired.

The Converse Direction ($\Leftarrow$). The converse direction can be shown by repeating the above discussion in reverse order. Assume that $\mathbf{e}_1 \in \mathrm{span}(\mathbf{L}_I^\top)$. Then there exists $\mathbf{u}' \in \mathbb{Z}_p^{|I|}$ such that $\mathbf{u}'^\top \mathbf{L}_I = \mathbf{e}_1^\top$. We extend $\mathbf{u}'$ to define $\mathbf{u}'' \in \mathbb{Z}_p^{\ell_0+\ell_1}$ so that $\mathbf{u}''_I = \mathbf{u}'$ and $\mathbf{u}''[i] = 0$ for $i \notin I$ hold. Here, $\mathbf{u}''_I \in \mathbb{Z}_p^{|I|}$ is a subvector of $\mathbf{u}''$ which is obtained by deleting all elements $\mathbf{u}''[i]$ for $i \notin I$. These conditions completely determine $\mathbf{u}''$. We denote this $\mathbf{u}''$ as $\mathbf{u}''^\top = (\mathbf{u}_0^\top, \mathbf{u}_1^\top)$ using $\mathbf{u}_0 \in \mathbb{Z}_p^{\ell_0}$ and $\mathbf{u}_1 \in \mathbb{Z}_p^{\ell_1}$. We note that $\mathbf{u}_0^\top \mathbf{L}_0 + \mathbf{u}_1^\top \mathbf{L}_1 = \mathbf{e}_1^\top$ holds by the definition.

Next we define $\mathbf{v}_i$ for $i \in [\bar{\ell}]$ as $\mathbf{v}_i = \mathbf{u}_0[i] \cdot \mathbf{p}(\rho(i))$ if $i \in [\ell_0]$ and $\mathbf{v}_i = \mathbf{0}_{\bar{k}+1}$ if $i \in [\ell_0 + 1, \bar{\ell}]$. We claim that $\langle \mathbf{v}_i, \mathbf{q}_S \rangle = 0$ holds for $i \in [\bar{\ell}]$. Here, we prove this. The case for $i \in [\ell_0 + 1, \bar{\ell}]$ is trivial. For the case of $i \in [1, \ell_0]$, we have

$$\langle \mathbf{v}_i, \mathbf{q}_S \rangle = \mathbf{u}_0[i] \cdot \langle \mathbf{p}(\rho(i)), \mathbf{q}_S \rangle = \mathbf{u}_0[i] \cdot Q_S(\rho(i)) = 0.$$

The last equation above holds because we have $Q_S(\rho(i)) = 0$ if $i \in I$ and $\mathbf{u}_0[i] = 0$ otherwise, by the definition of $\mathbf{u}_0[i]$.

We define $\mathbf{u}_2[i] \in \mathbb{Z}_p$ for $i \in [1, \ell_1]$ as $\mathbf{u}_2[i] = \mathbf{u}_1[i] / Q_S(\rho(\ell_0 + i))$ if $\mathbf{u}_1[i] \neq 0$ and $\mathbf{u}_2[i] = 0$ if $\mathbf{u}_1[i] = 0$. We have to show that $\mathbf{u}_2[i]$ are well defined by showing that $Q_S(\rho(\ell_0 + i)) \neq 0$ if $\mathbf{u}_1[i] \neq 0$ (i.e., division by 0 does not occur). If $\mathbf{u}_1[i] \neq 0$, then $(\ell_0 + i) \in I$ by the definition of $\mathbf{u}_1$. It implies that $(\rho(\ell_0 + i) = \neg \mathsf{Att}) \wedge (\mathsf{Att} \notin S)$ for some $\mathsf{Att} \in \mathbb{Z}_p$ and thus $Q_S(\rho(\ell_0 + i)) = \prod_{\omega \in S} (\mathsf{Att} - \omega) \neq 0$ holds as desired.

We also define $\mathbf{w}_i$ as $\mathbf{w}_i = \mathbf{u}_2[i] \cdot \mathbf{p}(\rho(\ell_0 + i))$ for $i \in [1, \ell_1]$ and $\mathbf{w}_i = \mathbf{0}_{\bar{k}+1}$ for $i \in [\ell_1 + 1, \bar{\ell}]$. Then, we have

$$\langle \mathbf{w}_i, \mathbf{q}_S \rangle = \mathbf{u}_2[i] \cdot \langle \mathbf{p}(\rho(\ell_0 + i)), \mathbf{q}_S \rangle = \mathbf{u}_2[i] \cdot Q_S(\rho(\ell_0 + i)) = \mathbf{u}_1[i]$$

for $i \in [1, \ell_1]$ and $\langle \mathbf{w}_i, \mathbf{q}_S \rangle = 0$ for $i \in [\ell_1 + 1, \bar{\ell}]$.

Finally, we define $\mathbf{u}$ and $\mathbf{v}$ as $\mathbf{u}^\top = (\mathbf{u}_0^\top, \mathbf{u}_1^\top, \mathbf{u}_2^\top)$ and $\mathbf{v}^\top = (\mathbf{v}_1^\top, \ldots, \mathbf{v}_{\bar{\ell}}^\top, \mathbf{w}_1^\top, \ldots, \mathbf{w}_{\bar{\ell}}^\top)$. Then, Equation (4) and (5) hold. By the properties of $\mathbf{u}$ and $\mathbf{v}$ we investigated so far, it is straightforward to see that $\mathbf{x}_0^\top + \mathbf{u}^\top \mathbf{X}^\top = \mathbf{y}_0^\top + \mathbf{v}^\top \mathbf{Y}$ holds. This concludes the proof of the claim.

This concludes the proof of the theorem.

5 From KP(CP)-ABE to KASP(CASP)

In this section, we show that monotonic KP-ABE with small universe (without bounds on the size of span programs) can be converted into KASP. We note that we can also obtain CP-ABE-to-CASP conversion by simply swapping key and ciphertext attribute.

5.1 The Conversion

We show how to construct KASP for dimension n from monotonic KP-ABE for parameter $N = (n\kappa + 1, -, -, -)$ and the size of attribute universe is $|\mathcal{U}| = 2n\kappa + 1$. Here, $\kappa = \lceil \log_2 p \rceil$. Similarly to Section 3, it suffices to define appropriate maps $f_{\mathsf{p}}^{\mathsf{KASP}\to\mathsf{KP}}$, $f_{\mathsf{e}}^{\mathsf{KASP}\to\mathsf{KP}}$, and $f_{\mathsf{k}}^{\mathsf{KASP}\to\mathsf{KP}}$. We define $f_{\mathsf{p}}^{\mathsf{KASP}\to\mathsf{KP}}(n) = N$. We set universe of attributes as $\mathcal{U} = \big\{\mathsf{Att}[i][j][b] \mid (i,j,b) \in [1,n] \times [1,\kappa] \times \{0,1\}\big\} \cup \{\mathsf{D}\}$. Intuitively, $\mathsf{Att}[i][j][b]$ represents "j-th least significant bit of binary representation of i-th element of a vector $\mathbf{x}$ is $b \in \{0,1\}$". D is a dummy attribute which will be assigned for all ciphertexts. For $\mathbf{x} \in \mathbb{Z}_p^n$, we define a set $S = f_{\mathsf{e}}^{\mathsf{KASP}\to\mathsf{KP}}(\mathbf{x})$ as

$$S = \left\{\mathsf{Att}[i][j][b] \mid (i,j) \in [1,n] \times [1,\kappa],\ b = \mathbf{x}[i][j]\right\} \cup \{\mathsf{D}\}.$$

In the above, we define $\mathbf{x}[i][j] \in \{0,1\}$ so that they satisfy $\mathbf{x}[i] = \sum_{j=1}^{\kappa} 2^{j-1} \cdot \mathbf{x}[i][j]$. Namely, $\mathbf{x}[i][j]$ is the j-th least significant bit of the binary representation of $\mathbf{x}[i] \in \mathbb{Z}_p$.

For an arithmetic span program $(\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_\ell) \in \mathbb{Z}_p^{m\times\ell}, \mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_\ell) \in \mathbb{Z}_p^{m\times\ell}, \rho)$ such that $\mathbf{Y}, \mathbf{Z} \in \mathbb{Z}_p^{m\times\ell}$, we define $f_{\mathsf{k}}^{\mathsf{KASP}\to\mathsf{KP}}(\mathbf{Y}, \mathbf{Z}, \rho) = (\mathbf{L}, \rho')$ as follows. At first, $\mathbf{L}$ is defined as

$$\mathbf{L} = \begin{pmatrix} \mathbf{G}_1 & \mathbf{J} & & \\ \mathbf{G}_2 & & \mathbf{J} & \\ \vdots & & & \ddots \\ \mathbf{G}_\ell & & & & \mathbf{J} \end{pmatrix} \in \mathbb{Z}_p^{\left((2\kappa+1)\ell\right)\times\left(\kappa\ell+m\right)}. \tag{6}$$

In the above, the matrix $\mathbf{J} \in \mathbb{Z}_p^{(2\kappa+1)\times\kappa}$ is defined as in Equation (2) (by setting $n = 1$) while $\mathbf{G}_i$ is defined as

$$\mathbf{G}_i = [\mathbf{g} \cdot \mathbf{y}_i^\top; \mathbf{z}_i^\top] = (\mathbf{0}_m, \mathbf{y}_i, \mathbf{0}_m, 2\mathbf{y}_i, \cdots, \mathbf{0}_m, 2^{\kappa-1}\mathbf{y}_i, \mathbf{z}_i)^\top \in \mathbb{Z}_p^{(2\kappa+1)\times m}$$

where $\mathbf{g} = (0, 1, 0, 2, \ldots, 0, 2^i, \ldots, 0, 2^{\kappa-1})^\top \in \mathbb{Z}_p^{2\kappa}$. We also define $\rho' : [(2\kappa+1)\ell] \to \mathcal{U}$ as follows. At first, if $i = 0 \mod (2\kappa+1)$, we set $\rho'(i) = \mathsf{D}$. Otherwise, i is represented as $i = (2\kappa+1)i' + 2j' + b' + 1$ using unique $i' \in [0, \ell-1]$, $j' \in [0, \kappa-1]$, and $b' \in \{0,1\}$. We set $\rho'(i) = \mathsf{Att}[\rho(i'+1)][j'+1][b']$.

Intuition. S can be seen as a binary representation of the information of $\mathbf{x}$. In the span program $(\mathbf{L}, \rho')$, $\mathbf{J}$ is used to constrain the form of linear combination among rows to a certain form. $\mathbf{G}_i$ as well as ρ', along with the above restriction, are designed so that linear combination of rows of $\mathbf{G}_i$ only can be a scalar multiple of the vector $(\mathbf{x}[\rho(i)]\mathbf{y}_i + \mathbf{z}_i)^\top$. Therefore, $(\mathbf{L}, \rho')$ essentially works as an arithmetic span program.

5.2 Correctness of the Conversion

By combining the following theorem and Lemma 1, we can see that KP-ABE for $N = (n\kappa + 1, -, -, -)$ can be converted into KASP for dimension n. The proof of the theorem appears in Appendix C.

Theorem 3. *For any* $\mathbf{x} \in \mathbb{Z}_p^n$, $\mathbf{Y} \in \mathbb{Z}_p^{m\times\ell}$, $\mathbf{Z} \in \mathbb{Z}_p^{m\times\ell}$, *and* $\rho : [\ell] \to [n]$, *it holds that*

$$R^{\mathsf{KP}}(S, (\mathbf{L}, \rho')) = 1 \Leftrightarrow R^{\mathsf{KASP}}(\mathbf{x}, (\mathbf{Y}, \mathbf{Z}, \rho)) = 1$$

where $S = f_{\mathsf{e}}^{\mathsf{KASP}\to\mathsf{KP}}(\mathbf{x})$ *and* $(\mathbf{L}, \rho') = f_{\mathsf{k}}^{\mathsf{KASP}\to\mathsf{KP}}(\mathbf{Y}, \mathbf{Z}, \rho)$ *is defined as above.*

6 Implications of Our Result

In this section, we discuss consequences of our results.

Equivalence between (bounded) ABE and DSE. We have shown that monotonic KP/CP-ABE for $(\bar{k}, \bar{\ell}, \bar{m}, \varphi)$ implies DSE (without delegation) in Section 3 and DSE implies non-monotonic KP/CP-ABE with large universe for $(\bar{k}, \bar{\ell}, \bar{m}, \varphi)$ in Section 4. Since non-monotonic KP/CP-ABE with large universe for $(\bar{k}, \bar{\ell}, \bar{m}, \varphi)$ trivially implies monotonic KP/CP-ABE with small universe for $(\bar{k}, \bar{\ell}, \bar{m}, \varphi)$, our results indicate that these PE schemes are essentially equivalent in the sense that they imply each other.

Equivalence between K(C)ASP and KP(CP)-ABE. Next, we consider the case where there is no restriction on the size of span programs. In Section 5, we showed that monotonic KP-ABE for $((\bar{k}+1)\kappa, -, -, -)$ implies KASP for $(\bar{k}, -, -, -)$. In Appendix E, we also show the converse direction. That is, we show that KASP for $(\bar{k} + 1, -, -, -)$ implies non-monotonic KP-ABE for $(\bar{k}, -, -, -)$ with large universe. Since non-monotonic KP-ABE for $(\bar{k}, -, -, -)$ trivially implies monotonic KP-ABE for $(\bar{k}, -, -, -)$, our results indicate that these PE schemes are essentially equivalent similarly to the above case. Similar implications hold for CP-ABE. See figure 1 for the overview.

By applying the conversions to existing schemes, we obtain various new schemes. The overviews of properties of resulting schemes and comparison with existing schemes are provided in Table 1, 2, and 3. All schemes in the tables are constructed in pairing groups. In the tables, we count the number of group elements to measure the size of master public keys ($|\mathsf{mpk}|$), ciphertexts ($|C|$), and private keys ($|\mathsf{sk}|$). Note that our conversions only can be applied to ABE schemes supporting span programs over $\mathbb{Z}_p$. Therefore, for ABE schemes constructed on composite order groups [28,1], our conversions are not applicable since they support span programs over $\mathbb{Z}_N$ where N is a product of several large primes. Similar restrictions are posed on DSE and K(C)ASP. Though it is quite plausible that our conversions work even in such cases assuming hardness of factoring N, we do not prove this in this paper.

New DSE Schemes. By applying our KP(CP)-ABE-to-DSE conversion to existing KP(CP)-ABE schemes, we obtain many new DSE schemes. Table 1 shows overview of obtained schemes. Specifically,

- From the unbounded KP-ABE schemes [34,36], we obtain the first DSE scheme with constant-size master public key (without delegation). Note that all previous schemes [23,12,14] require at least $O(n)$ group elements in master public key where n is the dimension of the scheme.
- From KP-ABE scheme with constant-size ciphertexts [2,25,38], we obtain the first DSE scheme with constant-size ciphertexts. All previous schemes [23,12,14] require at least $O(d_1)$ group elements in ciphertexts where d_1 is the dimension of the affine space associated to a ciphertext.

The schemes obtained from [34] achieves adaptive security. Furthemore, for schemes obtained from [36,2,25], we can define key delegation algorithm. The details of the key delegation algorithm will be given in Appendix D.

CP-ABE with Constant-Size Ciphertexts. Furthermore, by applying our DSE-to-non-monotonic-CP-ABE conversion in Section 4 to the DSE scheme with constant-size ciphertexts obtained above, we obtain the first non-monotonic CP-ABE with constant-size ciphertexts. Previous CP-ABE schemes with constant-size ciphertexts [16,11,10] only support threshold or more limited predicates[4]. See Table 2 for the overview.

New KASP and CASP Schemes. By applying the KP(CP)-ABE-to-K(C)ASP conversion in Section 5, we obtain many new K(C)ASP schemes. See Table 3 for the overview. Specifically,

- From the unbounded KP-ABE scheme [36], we obtain the first KASP scheme with constant-size master public key.
- From KP-ABE scheme with constant-size ciphertexts [2,25,38], we obtain the first KASP schemes with constant-size ciphertexts.
- From adaptively secure KP-ABE schemes [31,1] we obtain the first adaptively secure KASP scheme.
- From CP-ABE schemes [40,31,36], we also obtain several CASP schemes. These are the first CASP schemes. Among them, the construction obtained from [31] achieve adaptive security and the construction obtained from unbounded CP-ABE scheme [36] achieves constant-size master public key.

Note that only KASP scheme in the literature [26] is selectively secure and the master public key and ciphertext size are linear in the dimension of the scheme. We remark that the conversion is not applicable for schemes in [33,34] since these schemes are KP-ABE for $(*,*,*,\varphi)$ where φ is some polynomial, whereas our conversion requires the last parameter to be unbounded.

Table 1. Comparison of DSE Schemes:
The column “Delegation” states whether the scheme supports key delegation algorithm or not. “Unclear” means that we do not know whether it is possible to define key delegation algorithm for the scheme or not.

Schemes	$\lvert \mathsf{mpk} \rvert$	$\lvert C \rvert$	$\lvert \mathsf{sk} \rvert$	Delegation	Security
Hamburg [23]	$O(n)$	$O(d_1)$	$O(d_2)$	✓	Selective
CW [14]	$O(n^2)$	$O(nd_1)$	$O(n)$	✓	Selective
CZF [12]	$O(n)$	$O(d_1)$	$O(d_2)$	✓	Adaptive
Sec. 3 + [36]	$O(1)$	$O(nd_1\kappa)$	$O(n^2\kappa)$	✓	Selective
Sec. 3 + [34]	$O(1)$	$O(n^2d_1\kappa)$	$O(n^2\kappa)$	Unclear	Adaptive
Sec. 3 + [2]	$O(n^2\kappa)$	$O(1)$	$O(n^4\kappa^2)$	✓	Selective

[4] One would be able to obtain CP-ABE with constant-size ciphertexts supporting threshold formulae by applying the generic conversion in [20] to a KP-ABE scheme proposed in [2]. However, the resulting scheme supports more limited predicate compared to ours. To the best of our knowledge, this observation has not appeared elsewhere.

Table 2. Comparison of CP-ABE Schemes
We list previous CP-ABE schemes with compact ciphertexts in the table. We also include several state-of-the-art CP-ABE schemes for the sake of comparison.

Schemes	Expressiveness			Efficiency			Security
	Size of Universe	Monotonic or Non Monotonic	Access Policy	$\lvert\mathsf{mpk}\rvert$	$\lvert C\rvert$	$\lvert\mathsf{sk}\rvert$	
OT [34]	Large	Non-monotonic	Span Program	$O(1)$	$O(\ell)$	$O(k\varphi)$	Adaptive
AY [3]	Large	Monotonic	Span Program	$O(1)$	$O(\ell)$	$O(k)$	Adaptive
EMN+ [16]	Small	Monotonic	$(\bar{k}, \bar{k})$-threshold	$O(\bar{k})$	$O(1)$	$O(\bar{k})$	Selective
CZF [11]	Small	Non-Monotonic	AND-Gate	$O(\bar{k})$	$O(1)$	$O(\bar{k}^2)$	Selective
CCL+ [10]	Small	Monotonic	Threshold	$O(\bar{k})$	$O(1)$	$O(\bar{k}^2)$	Adaptive
Sec. 3,4 + [2]	Large	Non-monotonic	Span Program	$O((\bar{k}\bar{\ell})^2\kappa)$	$O(1)$	$O((\bar{k}\bar{\ell})^4\kappa^2)$	Selective

Table 3. Comparison of KASP and CASP Schemes

Schemes	Type	$\lvert\mathsf{mpk}\rvert$	$\lvert C\rvert$	$\lvert\mathsf{sk}\rvert$	Security
IW [26]	KASP	$O(n)$	$O(n)$	$O(\ell)$	Selective
Sec. 5 + [31]	KASP	$O(n\kappa)$	$O(n\kappa)$	$O(\ell\kappa)$	Adaptive
Sec. 5 + [2]	KASP	$O(n\kappa)$	$O(1)$	$O(\ell n\kappa^2)$	Selective
Sec. 5 + [36]	KASP	$O(1)$	$O(n\kappa)$	$O(\ell\kappa)$	Selective
Sec. 5 + [31]	CASP	$O(n\kappa)$	$O(\ell\kappa)$	$O(n\kappa)$	Adaptive
Sec. 5 + [36]	CASP	$O(1)$	$O(\ell\kappa)$	$O(n\kappa)$	Selective

7 Application to Attribute-Based Signature

Here, we discuss that our techniques developed in previous sections are also applicable to construct an attribute-based signatures (ABS) [32,33]. ABS is an advanced form of signature and can be considered as a signature analogue of ABE. In particular, it resembles CP-ABE in the sense that a private key is associated with a set of attributes while a signature is associated with a policy and a message. A user can sign on a message with a policy if and only if she has a private key associated with a set satisfying the policy. Roughly speaking, this property corresponds to the correctness and unforgeability. For ABS, we also require privacy. That is, we require that one cannot obtain any information about the attribute of the signer from a signature.

The construction of expressive ABS scheme with constant-size signatures has been open. All previous ABS schemes with constant-size signatures [24,10] only supports threshold predicates. The difficulty of constructing ABS with constant-size signatures seems to be related to the difficulty of construction of CP-ABE with constant-size ciphertexts. That is, it is hard to set constant number of group elements so that they include very complex information such as span programs.

To solve the problem, we first define the notion of predicate signature (PS) that is a signature analogue of PE. Then we construct a PS scheme that is dual of ABS: a private key is associated with a policy and a signature with a set. The scheme achieves constant-size signatures. This is not difficult to achieve because the signature is associated with a set which is a simpler object compared to a policy. The scheme is based on PS scheme for threshold predicate with constant-size signatures by [24]. We change the scheme mainly in two ways. At first, instead of using Shamir's secret sharing scheme, we use linear secret sharing scheme so that they support more general predicate. We also add some modification so that the signature size be even shorter. The signatures of the resulting scheme only consist of two group elements.

Since signature analogue of Lemma 1 holds (Lemma 2 in Appendix F), we can apply KP-ABE-to-non-monotonic-CP-ABE conversion (combination of the results in Section 3 and 4) to obtain the first ABS scheme with constant-size signatures supporting non-monotonic span program. See Appendix F for the details.

Acknowledgement. We would like to thank anonymous reviewers of ASACRYPT 2014 and TCC 2015 for their useful comments.

References

1. Nuttapong Attrapadung. Dual system encryption via doubly selective security: framework, fully secure functional encryption for regular languages, and More. In *EUROCRYPT*, pages 557–577, 2014.
2. Nuttapong Attrapadung, Benoît Libert, and Elie de Panafieu. Expressive key-policy attribute-based encryption with constant-size ciphertexts. In *Public Key Cryptography*, pages 90–108, 2011.
3. Nuttapong Attrapadung and Shota Yamada. Duality in ABE: Converting Attribute Based Encryption for Dual Predicate and Dual Policy via Computational Encodings. In *CT-RSA*, pages 87–105, 2015.
4. Amos Beimel. *Secure Schemes for Secret Sharing and Key Distribution*. PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1986.
5. Aleksandrs Belovs. Span-program-based quantum algorithm for the rank problem. Technical Report arXiv:1103.0842, arXiv.org, 2011. Available from `http://arxiv.org/abs/1103.0842`.
6. Amos Beimel, and Yuval Ishai. On the Power of Nonlinear Secret-Sharing. *IEEE Conference on Computational Complexity*, pages 188–202, 2001.
7. 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*, pages 533–556, 2014.
8. Dan Boneh and Michael Hamburg. Generalized identity based and broadcast encryption schemes. In *ASIACRYPT*, pages 455–470, 2008.
9. Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In *TCC*, pages 253–273, 2011.
10. Cheng Chen, Jie Chen, Hoon Wei Lim, Zhenfeng Zhang, Dengguo Feng, San Ling, and Huaxiong Wang. Fully secure attribute-based systems with short ciphertexts/signatures and threshold access structures. In *CT-RSA*, pages 50–67, 2013.
11. Cheng Chen, Zhenfeng Zhang, and Dengguo Feng. Efficient ciphertext policy attribute-based encryption with constant-size ciphertext and constant computation-cost. In *ProvSec*, pages 84–101, 2011.
12. Cheng Chen, Zhenfeng Zhang, and Dengguo Feng. Fully secure doubly-spatial encryption under simple assumptions. In *ProvSec*, pages 253–263, 2012.

13. Jie Chen, Hoon Wei Lim, San Ling, and Huaxiong Wang. The relation and transformation between hierarchical inner product encryption and spatial encryption. *Des. Codes Cryptography*, 71(2):pages 347–364, 2014.
14. Jie Chen and Hoeteck Wee. Doubly spatial encryption from DBDH. *Theor. Comput. Sci*, 543:pages 79–89, 2014
15. Jie Chen and Hoeteck Wee. Semi-adaptive attribute-based encryption and improved delegation for Boolean formula. *SCN*, pages 277–297, 2014.
16. Keita Emura, Atsuko Miyaji, Akito Nomura, Kazumasa Omote, and Masakazu Soshi. A ciphertext-policy attribute-based encryption scheme with constant ciphertext length. In *ISPEC*, pages 13–23, 2009.
17. Sanjam Garg, Craig Gentry, Shai Halevi, Amit Sahai, and Brent Waters. Attribute-based encryption for circuits from multilinear maps. In *CRYPTO (2)*, pages 479–499, 2013.
18. Aijun Ge, Rui Zhang, Cheng Chen, Chuangui Ma, and Zhenfeng Zhang. Threshold ciphertext policy attribute-based encryption with constant size ciphertexts. In *ACISP*, pages 336–349, 2012.
19. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In *STOC*, pages 545–554, 2013.
20. Vipul Goyal, Abhishek Jain, Omkant Pandey, and Amit Sahai. Bounded ciphertext policy attribute based encryption. In *ICALP (2)*, pages 579–591, 2008.
21. Vipul Goyal, Virendra Kumar, Satyanarayana V. Lokam, and Mohammad Mahmoody. On Black-Box Reductions between Predicate Encryption Schemes. In *TCC*, pages 440–457, 2012.
22. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM Conference on Computer and Communications Security*, pages 89–98, 2006.
23. Mike Hamburg. Spatial encryption. *IACR Cryptology ePrint Archive*, 2011:389, 2011.
24. Javier Herranz, Fabien Laguillaumie, Benoît Libert, and Carla Ràfols. Short attribute-based signatures for threshold predicates. In *CT-RSA*, pages 51–67, 2012.
25. Susan Hohenberger and Brent Waters. Attribute-based encryption with fast decryption. In *Public Key Cryptography*, pages 162–179, 2013.
26. Yuval Ishai and Hoeteck Wee. Partial Garbling and Their Applications. In *ICALP (1)*, pages 650–662, 2014.
27. Jonathan Katz, Amit Sahai, and Brent Waters. Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products. In *EUROCRYPT*, pages 146–162, 2008.
28. 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*, pages 62–91, 2010.
29. Allison B. Lewko and Brent Waters. Unbounded hibe and attribute-based encryption. In *EUROCRYPT*, pages 547–567, 2011.
30. Allison B. Lewko and Brent Waters. Decentralizing Attribute-Based Encryption. In *EUROCRYPT*, pages 568–588, 2011.
31. Allison B. Lewko and Brent Waters. New Proof Methods for Attribute-Based Encryption: Achieving Full Security through Selective Techniques. In *CRYPTO*, pages 180–198, 2012.
32. Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek. Attribute-Based Signatures. In *CT-RSA*, pages 376–392, 2011.
33. Tatsuaki Okamoto and Katsuyuki Takashima. Efficient attribute-based signatures for non-monotone predicates in the standard model In *PKC*, pages 35–52, 2011.
34. Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure unbounded inner-product and attribute-based encryption. In *ASIACRYPT*, pages 349–366, 2012.

35. Bryan Parno, Mariana Raykova, and Vinod Vaikuntanathan. How to Delegate and Verify in Public: Verifiable Computation from Attribute-Based Encryption. In *TCC*, pages 422–439, 2012.
36. Yannis Rouselakis and Brent Waters. Practical constructions and new proof methods for large universe attribute-based encryption. In *ACM Conference on Computer and Communications Security*, pages 463–474, 2013.
37. Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, pages 457–473, 2005.
38. Katsuyuki Takashima. Expressive attribute-based encryption with constant-size ciphertexts from the decisional linear assumption. *SCN*, pages 298–317, 2014.
39. Brent Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, pages 114–127, 2005.
40. Brent Waters. Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. In *PKC*, pages 53–70, 2011.
41. Brent Waters. Functional Encryption for Regular Languages. In *CRYPTO*, pages 218–235, 2012.
42. Shota Yamada, Nuttapong Attrapadung, Goichiro Hanaoka, and Noboru Kunihiro. A framework and compact constructions for non-monotonic attribute-based encryption. In *Public Key Cryptography*, pages 275–292, 2014.

A Proof of Lemma 1

Proof. We prove the lemma for the case of adaptive security. The same proof works for other cases. We construct an adversary $\mathcal{B}$ against Π from an adversary $\mathcal{A}$ against Π' so that $\mathcal{B}$ has the same advantage as $\mathcal{A}$ and the running time of $\mathcal{B}$ is only polynomially larger that $\mathcal{A}$. At first, $\mathsf{Setup}(\lambda, f_{\mathsf{p}}(N')) = \mathsf{Setup}'(\lambda, N') \to \mathsf{mpk}$ is run and mpk is given to $\mathcal{B}$. $\mathcal{B}$ gives mpk to $\mathcal{A}$. In $\mathbf{Phase1}$, when $\mathcal{A}$ makes key query for $Y' \in B'_{N'}$, $\mathcal{B}$ computes $Y = f_{\mathsf{k}}(Y')$ and makes a key-extraction query for its challenger. Then, $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, f_{\mathsf{k}}(Y')) = \mathsf{KeyGen}'(\mathsf{msk}, \mathsf{mpk}, Y') \to \mathsf{sk}_{Y'}$ is run and $\mathcal{B}$ is given $\mathsf{sk}_{Y'}$. Then, $\mathcal{B}$ gives it to $\mathcal{A}$. At some point, $\mathcal{A}$ outputs two equal length messages M_0 and M_1 and challenge ciphertext attribute $X'^{\star} \in A'_{N'}$. Then $\mathcal{B}$ computes $X^{\star} = f_{\mathsf{e}}(X'^{\star})$ and submits $X^{\star}$, M_0, and M_1 to its challenger. Then, $\mathsf{Encrypt}(\mathsf{mpk}, \mathsf{M}_\beta, f_{\mathsf{e}}(X'^{\star})) = \mathsf{Encrypt}'(\mathsf{mpk}, \mathsf{M}_\beta, X'^{\star}) \to C^{\star}$ is run and $C^{\star}$ is given to $\mathcal{B}$. Here, β is a random bit. $\mathcal{B}$ gives $C^{\star}$ to $\mathcal{A}$. In **Phase2**, $\mathcal{B}$ deals with key-extraction queries made by $\mathcal{A}$ as in **Phase1**. Finally, $\mathcal{B}$ outputs the same bit as $\mathcal{A}$. We check that $\mathcal{B}$ has not made any prohibited key-extraction query. This holds because we have $R^{\mathsf{F}}_{f_{\mathsf{p}}(N')}(f_{\mathsf{e}}(X'^{\star}), f_{\mathsf{k}}(Y')) = R^{\mathsf{F}'}_{N'}(X'^{\star}, Y') = 0$ for all Y' queried by $\mathcal{A}$. It is straightforward to see that the simulation by $\mathcal{B}$ is perfect and $\mathcal{B}$ has exactly the same advantage as $\mathcal{A}$.

B Proof of Theorem 1

Proof. Let $I \subset [1, (2n\kappa + 1)(n + 1) + d_2 + 1]$ be $I = \{i | \rho(i) \in S\}$. We define $\mathbf{L}_I$ which is the sub-matrix of $\mathbf{L}$ formed by rows whose index is in I. From the definition of $f_{\mathsf{e}}^{\mathsf{DSE}\to\mathsf{KP}}$, we have that $\mathbf{L}_I$ is in the form of

$$\mathbf{L}_I = \begin{pmatrix} \mathbf{e}_1 & \mathbf{e}_1 + \mathbf{e}_{d_2+2} & \mathbf{y}_0^\top & & & & & & \\ & & \mathbf{Y}^\top & & & & & & \\ & & \mathbf{E}_0 & \mathbf{J}' & & & & & \\ & & \mathbf{E}_1 & & \mathbf{J}' & & & & \\ & & \vdots & & & \ddots & & & \\ & & \mathbf{E}_{d_1} & & & & \mathbf{J}' & & \\ & & & & & & & \mathbf{1}_{n\kappa}^\top & \\ & & & & & & & & \ddots \\ & & & & & & & & & \mathbf{1}_{n\kappa}^\top \end{pmatrix} \in \mathbb{Z}_p^{\big((n\kappa+1)(d_1+1)+n-d_1+d_2+1\big)\times\big((n\kappa+1)(n+1)+1\big)}$$

where

$$\mathbf{E}_i = \begin{pmatrix} \mathbf{g}_{i,1} & & & \\ & \mathbf{g}_{i,2} & & \\ & & \ddots & \\ & & & \mathbf{g}_{i,n} \\ 0 & 0 & \dots & 0 \end{pmatrix} \in \mathbb{Z}_p^{(n\kappa+1)\times n}, \qquad \mathbf{J}' = \begin{pmatrix} -1 & & & \\ & -1 & & \\ & & \ddots & \\ & & & -1 \\ 1 & 1 & \dots & 1 \end{pmatrix} \in \mathbb{Z}_p^{(n\kappa+1)\times n\kappa}.$$

for $i \in [0, d_1]$. In the above, $\mathbf{g}_{i,j}$ is

$$\mathbf{g}_{i,j} = (\mathbf{x}_i[j][1], 2\mathbf{x}_i[j][2], \dots, 2^{\kappa-1}\mathbf{x}_i[j][\kappa])^\top \in \mathbb{Z}_p^\kappa.$$

We remark that it holds that $\langle \mathbf{1}_\kappa, \mathbf{g}_{i,j}\rangle = \mathbf{x}_i[j]$ by the definition of $\mathbf{x}_i[j][k]$ and thus $\mathbf{E}_i^\top \cdot \mathbf{1}_{n\kappa+1} = \mathbf{x}_i$ holds. We also remark that if $\mathbf{v}^\top\mathbf{J}' = \mathbf{0}$ holds for some $\mathbf{v} \in \mathbb{Z}_p^{n\kappa+1}$, then there exists $v \in \mathbb{Z}_p$ such that $\mathbf{v} = v\mathbf{1}_{n\kappa+1}$. These properties will be used later.

By the definition of R^{KP} and R^{DSE} it suffices to show the following claim.

Claim. $\mathbf{e}_1 \in \mathrm{span}(\mathbf{L}_I^\top) \Leftrightarrow \big(\mathbf{x}_0 + \mathrm{span}(\mathbf{X})\big) \cap \big(\mathbf{y}_0 + \mathrm{span}(\mathbf{Y})\big) \neq \emptyset$.

Proof ((Proof of the claim)). We show both direction separately.

The Forward Direction ($\Rightarrow$). Let us assume that $\mathbf{e}_1 \in \mathrm{span}(\mathbf{L}_I^\top)$. By the assumption, there exists $\mathbf{u} \in \mathbb{Z}_p^{\big((n\kappa+1)(d_1+1)+n-d_1+d_2+1\big)}$ such that $\mathbf{u}^\top\mathbf{L}_I = \mathbf{e}_1^\top$. We write this $\mathbf{u}$ as

$$\mathbf{u}^\top = \big(\underbrace{v}_{1}, \underbrace{\mathbf{v}^\top}_{d_2}, \underbrace{\mathbf{u}_0^\top}_{n\kappa+1}, \underbrace{\mathbf{u}_1^\top}_{n\kappa+1}, \dots, \underbrace{\mathbf{u}_{d_1}^\top}_{n\kappa+1}, \underbrace{u_{d_1+1}}_{1}, \dots, \underbrace{u_n}_{1}\big).$$

Then,

$$\begin{aligned} \mathbf{u}^\top \cdot \mathbf{L}_I &= \Big(v, v + \langle \mathbf{u}_0, \mathbf{e}_1\rangle, v\mathbf{y}_0^\top + \mathbf{v}^\top\mathbf{Y}^\top + \sum_{i=0}^{d_1} \mathbf{u}_i^\top\mathbf{E}_i, \mathbf{u}_0^\top \cdot \mathbf{J}', \dots, \mathbf{u}_{d_1}^\top \cdot \mathbf{J}', u_{d_1+1}\mathbf{1}_{n\kappa+1}^\top, \dots, u_n\mathbf{1}_{n\kappa+1}^\top\Big) \\ &= \mathbf{e}_1^\top. \end{aligned}$$

By comparing each element of the vector, it follows that $u_{d_1+1} = \cdots = u_n = 0$. Furthermore, since $\mathbf{u}_i^\top \cdot \mathbf{J}' = \mathbf{0}$ for $i \in [0, d_1]$, there exist $\{u_i \in \mathbb{Z}_p\}_{i\in[0,d_1]}$ such that $\mathbf{u}_i = u_i \mathbf{1}_{n\kappa+1}$. By comparing the first and the second element of the vector, we obtain $v = 1$ and $v + \langle \mathbf{u}_0, \mathbf{e}_1\rangle = 1 + u_0\langle \mathbf{1}^\top, \mathbf{e}_{n\kappa+1}\rangle = 1 + u_0 = 0 \Leftrightarrow u_0 = -1$. Finally, we have that $\sum_{i=0}^{d_1} \mathbf{u}_i^\top \mathbf{E}_i + v\mathbf{y}_0^\top + \mathbf{v}^\top \mathbf{Y}^\top = \mathbf{0}$ and thus

$$-\sum_{i=0}^{d_1} \mathbf{E}_i^\top \mathbf{u}_i = \mathbf{y}_0 + \mathbf{Y} \cdot \mathbf{v}.$$

Left hand side of the equation is

$$-\sum_{i=0}^{d_1} \mathbf{E}_i^\top \mathbf{u}_i = -u_0 \mathbf{E}_0^\top \cdot \mathbf{1}_{n\kappa+1} - \sum_{i=1}^{d_1} u_i \mathbf{E}_i^\top \cdot \mathbf{1}_{n\kappa+1} = \mathbf{x}_0 - \sum_{i=1}^{d_1} u_i \cdot \mathbf{x}_i \in \big(\mathbf{x}_0 + \mathrm{span}(\mathbf{X})\big)$$

while right hand side is $\mathbf{y}_0 + \mathbf{Y}\cdot(-\mathbf{v}) \in (\mathbf{y}_0 + \mathrm{span}(\mathbf{Y}))$. This implies that $\big(\mathbf{x}_0 + \mathrm{span}(\mathbf{X})\big) \cap \big(\mathbf{y}_0 + \mathrm{span}(\mathbf{Y})\big) \neq \emptyset$.

The Converse Direction ($\Leftarrow$). Let us assume that $\big(\mathbf{x}_0 + \mathrm{span}(\mathbf{X})\big) \cap \big(\mathbf{y}_0 + \mathrm{span}(\mathbf{Y})\big) \neq \emptyset$. Then, there exist $\{u_i \in \mathbb{Z}_p\}_{i\in[1,d_1]}$ and $\{v_i \in \mathbb{Z}_p\}_{i\in[1,d_2]}$ such that $\mathbf{x}_0 + \sum_{i=1}^{d_1} u_i \mathbf{x}_i = \mathbf{y}_0 + \sum_{i=1}^{d_2} v_i \mathbf{y}_i$. We set a vector $\mathbf{u}$ as

$$\mathbf{u}^\top = (1, \underbrace{v_1, \ldots, v_{d_2}}_{d_2} \underbrace{-\mathbf{1}_{n\kappa+1}^\top, -u_1\mathbf{1}_{n\kappa+1}^\top, \ldots, -u_{d_1}\mathbf{1}_{n\kappa+1}^\top}_{(n\kappa+1)(d_1+1)}, \underbrace{0, \ldots, 0}_{n-d_1})).$$

Then we have

$$\begin{aligned}
\mathbf{u}^\top \mathbf{L}_I = (1, 1-1, \mathbf{y}_0^\top + \sum_{i=1}^{d_2} v_i \mathbf{y}_i^\top - \mathbf{1}_{n\kappa+1}^\top(\mathbf{E}_0 + \sum_{i=1}^{d_1} u_i \mathbf{E}_i),\\
-\mathbf{1}_{n\kappa+1}^\top \mathbf{J}', -u_1\mathbf{1}_{n\kappa+1}^\top \mathbf{J}', \ldots, -u_n \mathbf{1}_{n\kappa+1}^\top \mathbf{J}', 0\ldots, 0)\\
= (1, 0, \Big(\mathbf{y}_0^\top + \sum_{i=1}^{d_2} v_i \mathbf{y}_i^\top\Big) - \Big(\mathbf{x}_0^\top + \sum_{i=1}^{d_1} u_i \mathbf{x}_i^\top\Big), 0\ldots, 0) = \mathbf{e}_1^\top
\end{aligned}$$

as desired. This concludes the proof of the claim.

This concludes the proof of the theorem.

C Proof of Theorem 3

Proof. Let $I \subset [1, (2\kappa+1)\ell]$ be $I = \{i | \rho'(i) \in S\}$. We define $\mathbf{L}_I$ which is the submatrix of $\mathbf{L}$ formed by rows whose index is in I. From the definition of $f_{\mathsf{e}}^{\mathsf{KASP}\to\mathsf{KP}}$, we have that $\mathbf{L}_I$ is in the form of

$$\mathbf{L}_I = \begin{pmatrix} \mathbf{G}_1' & \mathbf{J}' & & \\ \mathbf{G}_2' & & \mathbf{J}' & \\ \vdots & & & \ddots \\ \mathbf{G}_\ell' & & & & \mathbf{J}' \end{pmatrix} \in \mathbb{Z}_p^{\big((\kappa+1)\ell\big)\times\big(\kappa\ell+m\big)}$$

where

$$\mathbf{G}'_i = [\mathbf{g}_i \cdot \mathbf{y}_i^\top ; \mathbf{z}_i^\top] \in \mathbb{Z}_p^{(\kappa+1)\times m} \qquad \mathbf{J}' = \begin{pmatrix} -1 & & & \\ & -1 & & \\ & & \ddots & \\ & & & -1 \\ 1 & 1 & \dots & 1 \end{pmatrix} \in \mathbb{Z}_p^{(\kappa+1)\times\kappa}.$$

In the above, $\mathbf{g}_i$ is

$$\mathbf{g}_i = (\mathbf{x}[\rho(i)][1], 2\mathbf{x}[\rho(i)][2], \dots, 2^{\kappa-1}\mathbf{x}[\rho(i)][\kappa])^\top \in \mathbb{Z}_p^\kappa.$$

We remark that it holds that $\langle \mathbf{1}_\kappa, \mathbf{g}_i \rangle = \mathbf{x}[\rho(i)]$ by the definition of $\mathbf{x}[\rho(i)][j]$ and thus $\mathbf{G}'^\top_i \cdot \mathbf{1}_{\kappa+1} = \mathbf{x}[\rho(i)]\mathbf{y}_i + \mathbf{z}_i$ holds. We also remark that if $\mathbf{v}^\top \mathbf{J}' = \mathbf{0}$ holds for some $\mathbf{v} \in \mathbb{Z}_p^{\kappa+1}$, then there exists $v \in \mathbb{Z}_p$ such that $\mathbf{v} = v\mathbf{1}_{\kappa+1}$. These properties will be used later.

By the definition of R^{KP} and R^{KASP}, it suffices to show the following claim.

Claim. $\mathbf{e}_1 \in \mathrm{span}(\mathbf{L}_I^\top) \Leftrightarrow \mathbf{e}_1 \in \mathrm{span}(\{\mathbf{x}[\rho(i)]\mathbf{y}_i + \mathbf{z}_i\}_{i\in[\ell]})$.

Proof (Proof of the claim). We first show the forward direction ($\Rightarrow$). We assume that $\mathbf{e}_1 \in \mathrm{span}(\mathbf{L}_I^\top)$. By the assumption, there exists $\mathbf{u} \in \mathbb{Z}_p^{(\kappa+1)\ell}$ such that $\mathbf{u}^\top \mathbf{L}_I = \mathbf{e}_1^\top$. We write this $\mathbf{u}$ as

$$\mathbf{u}^\top = (\underbrace{\mathbf{u}_1^\top}_{\kappa+1}, \underbrace{\mathbf{u}_2^\top}_{\kappa+1}, \dots, \underbrace{\mathbf{u}_\ell^\top}_{\kappa+1}).$$

Then, we have that

$$\mathbf{e}_1^\top = \mathbf{u}^\top \cdot \mathbf{L}_I = \left(\sum_{i\in[\ell]} \mathbf{u}_i^\top \mathbf{G}'_i, \mathbf{u}_1^\top \mathbf{J}', \dots, \mathbf{u}_\ell^\top \mathbf{J}' \right).$$

Since $\mathbf{u}_i^\top \cdot \mathbf{J}' = \mathbf{0}$ for $i \in [\ell]$, there exist $\{u_i \in \mathbb{Z}_p\}_{i\in[\ell]}$ such that $\mathbf{u}_i = u_i \mathbf{1}_{\kappa+1}$. Then, we have

$$\mathbf{e}_1^\top = \sum_{i\in[\ell]} \mathbf{u}_i^\top \mathbf{G}'_i = \sum_{i\in[\ell]} u_i \mathbf{1}_{\kappa+1}^\top \mathbf{G}'_i = \sum_{i\in[\ell]} u_i (\mathbf{x}[\rho(i)] \cdot \mathbf{y}_i + \mathbf{z}_i)^\top.$$

This implies $\mathbf{e}_1 \in \mathrm{span}(\{\mathbf{x}[\rho(i)]\mathbf{y}_i + \mathbf{z}_i\}_{i\in[\ell]})$, as desired.

Next, we prove the other direction ($\Leftarrow$). We assume that $\mathbf{e}_1 \in \mathrm{span}(\{\mathbf{x}[\rho(i)]\mathbf{y}_i + \mathbf{z}_i\}_{i\in[\ell]})$. Then, there exist $\{u_i \in \mathbb{Z}_p\}_{i\in[\ell]}$ such that $\sum_{i\in[\ell]} u_i(\mathbf{x}[\rho(i)] \cdot \mathbf{y}_i + \mathbf{z}_i) = \mathbf{e}_1$. We set a vector $\mathbf{u} \in \mathbb{Z}_p^{(\kappa+1)\ell}$ as

$$\mathbf{u}^\top = (u_1 \mathbf{1}_{\kappa+1}^\top, \dots, u_\ell \mathbf{1}_{\kappa+1}^\top)$$

Then, we have that

$$\mathbf{u}^\top \cdot \mathbf{L}_I = \left(\sum_{i \in [\ell]} u_i \mathbf{1}_{\kappa+1}^\top \mathbf{G}'_i, u_1 \mathbf{1}_{\kappa+1}^\top \mathbf{J}', \ldots, \mathbf{u}_\ell \mathbf{1}_{\kappa+1}^\top \mathbf{J}' \right)$$
$$= \left(\sum_{i \in [\ell]} u_i (\mathbf{x}[\rho(i)] \mathbf{y}_i + \mathbf{z}_i)^\top, \mathbf{0}_\kappa^\top, \ldots, \mathbf{0}_\kappa^\top \right) = \mathbf{e}_1^\top.$$

This implies that $\mathbf{e}_1 \in \mathrm{span}(\mathbf{L}_I^\top)$, as desired. This concludes the proof of the claim.

This concludes the proof of the theorem.

D Key Delegation Algorithm

In Section 3, we showed that KP-ABE scheme can be converted into DSE scheme without delegation. The conversion is completely generic and can be applicable to many existing schemes. Here, we show that it is possible to add key delegation algorithm for resulting schemes in some cases. To capture such cases, we define the following template for monotonic KP-ABE. If the original KP-ABE scheme is an instance of the template, we can add key delegation algorithm to the DSE scheme obtained by our conversion.

Let $\mathbb{G}, \mathbb{G}_T$ be underlying bilinear groups of order p. The universe of attribute is denoted by $\mathcal{U}$.

$\mathsf{Setup}(\lambda, N)$: Given a security parameter $\lambda \in \mathbb{N}$ and some bounds on parameters $N \in \mathbb{Z}$, the algorithm selects bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ and a generator $g \xleftarrow{\$} \mathbb{G}$. It computes $e(g,g)^\alpha$ for a random $\alpha \xleftarrow{\$} \mathbb{Z}_p$ and sets parameters pp. We assume that g is included in pp. The master secret key consists of $\mathsf{msk} = \alpha$ while the master public key is $\mathsf{mpk} = (e(g,g)^\alpha, \mathsf{pp})$.

We assume that pp defines publicly computable functions $f_0 : \mathcal{U} \to \mathbb{G}^{\ell'}$ and $f_1 : \mathcal{U} \to \mathbb{G}$.

$\mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, (\mathbf{L}, \rho))$: The input to the algorithm is the master secret key msk, master public key mpk, and a monotone span program $(\mathbf{L}, \rho)$. Let $\mathbf{L}$ be an $\ell \times m$ matrix. First, it picks random $\mathbf{s}[2], \ldots, \mathbf{s}[m], r_1, \ldots, r_\ell \xleftarrow{\$} \mathbb{Z}_p$ and sets the vector $\mathbf{s} = (\alpha, \mathbf{s}[2], \ldots, \mathbf{s}[m])^\top$. Then, it computes shares of α for $\rho(i)$ by $\lambda_i = \mathbf{L}_i \cdot \mathbf{s}$ for $i \in [1, \ell]$. Here, $\mathbf{L}_i$ is i-th *row* of $\mathbf{L}$. It then outputs private key

$$\mathsf{sk}_{(\mathbf{L},\rho)} = \left\{ D_{i,0} = f_0(\rho(i))^{r_i}, \quad D_{i,1} = g^{\lambda_i} \cdot f_1(\rho(i))^{r_i} \right\}_{i \in [1,\ell]}.$$

Example. Many existing KP-ABE schemes such as [22,2,25,36] can be captured by the above template. For example, unbounded monotonic KP-ABE scheme in [36] can be captured by defining pp, f_0, f_1 as

$$\mathsf{pp} = (g, u, h, w), \quad f_0(\mathsf{Att}) = (u^{\mathsf{Att}} \cdot h,\ g), \quad f_1(\mathsf{Att}) = w.$$

Monotonic KP-ABE scheme with constant-size ciphertexts in [2] can be captured by defining pp, f_0, f_1 as

$$\mathsf{pp} = (g, \{h_i\}_{i\in[0,\bar{k}+1]}), \quad f_0(\mathsf{Att}) = (\{h_1^{-\mathsf{Att}^{i-1}} \cdot h_i\}_{i\in[2,\bar{k}+1]},\ g), \quad f_1(\mathsf{Att}) = h_0.$$

If the setup and key generation algorithm of KP-ABE scheme $\Pi = \{\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt}\}$ satisfy the template, one can add key delegation algorithm to a DSE scheme Π' that is obtained by applying our conversion in Section 3 to the KP-ABE scheme Π.

Before showing key delegation algorithm, we first define ReRand algorithm for Π that re-randomizes private key. This will be used as a subroutine in the key delegation algorithm of the DSE scheme Π'.

$\mathsf{ReRand}(\mathsf{mpk}, \mathsf{sk}_{(\mathbf{L},\rho)}, (\mathbf{L}, \rho))$: It takes as input mpk, a private key $\mathsf{sk}_{(\mathbf{L},\rho)}$, and a span program $(\mathbf{L}, \rho)$ and outputs re-randomized private key $\mathsf{sk}'_{(\mathbf{L},\rho)}$ for the same span program. Let $\mathbf{L}$ be $\ell \times m$ matrix. It first parse the private key as $\{D_{i,0}, D_{i,1}\}_{i\in[1,\ell]}$. Then it picks $\mathbf{s}'[2], \ldots, \mathbf{s}'[m], r'_1, \ldots, r'_\ell \xleftarrow{\$} \mathbb{Z}_p$ and sets $\lambda'_i = \mathbf{L}_i \cdot \mathbf{s}'$ for each $i \in [1, \ell]$ where $\mathbf{s}' = (0, \mathbf{s}'[2], \ldots, \mathbf{s}'[m])^\top$ and $\mathbf{L}_i$ is the i-th *row* of $\mathbf{L}$. Finally, it outputs

$$\{D'_{i,0} = D_{i,0} \cdot f_0(\rho(i))^{r'_i}, \quad D'_{i,1} = D_{i,1} \cdot g^{\lambda'_i} \cdot f_1(\rho(i))^{r'_i}\}_{i\in[1,\ell]}.$$

We have following claim.

Claim. For all λ, $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(\lambda)$, $(\mathbf{L}, \rho)$, and $\mathsf{sk}_{(\mathbf{L},\rho)} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, (\mathbf{L}, \rho))$, we have the following distributions are identical:

$$\mathsf{ReRand}(\mathsf{mpk}, \mathsf{sk}_{(\mathbf{L},\rho)}, (\mathbf{L}, \rho)) \approx \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, (\mathbf{L}, \rho)). \qquad (7)$$

Proof. We have that $\mathsf{sk}_{(\mathbf{L},\rho)}$ is in the form of $\mathsf{sk}_{(\mathbf{L},\rho)} = \{D_{i,0} = f_0(\rho(i))^{r_i}, D_{i,1} = g^{\lambda_i} \cdot f_1(\rho(i))^{r_i}\}_{i\in[1,\ell]}$ where $\lambda_i = \mathbf{L}_i \cdot \mathbf{s}$ for some $\mathbf{s} = (\alpha, \mathbf{s}[2], \ldots, \mathbf{s}[m])^\top \in \mathbb{Z}_p^m$ and $\{r_i\}_{i\in[1,\ell]}$. Then, the output of $\mathsf{ReRand}(\mathsf{mpk}, \mathsf{sk}_{(\mathbf{L},\rho)}, (\mathbf{L}, \rho))$ is $\{\tilde{D}_{i,0} = f_0(\rho(i))^{r_i+r'_i}$, $\tilde{D}_{i,1} = g^{\lambda_i+\lambda'_i} \cdot f_1(\rho(i))^{r_i+r'_i}\}_{i\in[1,\ell]}$. We have that $\lambda'_i + \lambda_i = \mathbf{L}_i \cdot (\mathbf{s}+\mathbf{s}')$ and $\mathbf{s}+\mathbf{s}' = (\alpha, \mathbf{s}[2] + \mathbf{s}'[2], \ldots, \mathbf{s}[m] + \mathbf{s}'[m])^\top$. Here, $\mathbf{s}[i] + \mathbf{s}'[i]$ for $i \in [2, m]$ and $r_i + r'_i$ for $i \in [1, m]$ are uniformly distributed over $\mathbb{Z}_p$ and independent from the randomness used to generate $\mathsf{sk}_{(\mathbf{L},\rho)}$ (namely, $\{\mathbf{s}[i]\}_{i\in[2,m]}$ and $\{r_i\}_{i\in[1,m]}$). Thus the output distribution of $\mathsf{ReRand}(\mathsf{mpk}, \mathsf{sk}_{(\mathbf{L},\rho)}, (\mathbf{L}, \rho))$ is the same as that of $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, (\mathbf{L}, \rho))$.

Now we define key delegation algorithm KeyDel' for Π'. In the following, $n' = (2n\kappa + 1)(n+1)$.

$\mathsf{KeyDel}'(\mathsf{mpk}, \mathsf{sk}_{(\mathbf{y}_0,\mathbf{Y})}, (\mathbf{y}_0, \mathbf{Y}), (\hat{\mathbf{y}}_0, \hat{\mathbf{Y}}))$: It takes as input the master public key mpk, a private key $\mathsf{sk}_{(\mathbf{y}_0,\mathbf{Y})}$ for $(\mathbf{y}_0, \mathbf{Y}) \in \mathbb{Z}_p^n \times \mathbb{Z}_p^{n\times d}$, and $(\hat{\mathbf{y}}_0, \hat{\mathbf{Y}}) \in \mathbb{Z}_p^n \times \mathbb{Z}_p^{n\times \hat{d}}$ such that $\hat{\mathbf{y}}_0 + \mathrm{span}(\hat{\mathbf{Y}}) \subseteq \mathbf{y}_0 + \mathrm{span}(\mathbf{Y})$. Here, $\mathsf{sk}_{(\mathbf{y}_0,\mathbf{Y})}$ is generated by $\mathsf{sk}_{(\mathbf{y}_0,\mathbf{Y})} \leftarrow \mathsf{KeyGen}'(\mathsf{msk}, \mathsf{mpk}, (\mathbf{y}_0, \mathbf{Y})) = \mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, (\mathbf{L}, \rho))$ for $(\mathbf{L}, \rho) = f_{\mathsf{k}}^{\mathsf{DSE}\to\mathsf{KP}}(\mathbf{y}_0, \mathbf{Y})$. The key delegation algorithm first parses the private key as $\mathsf{sk}_{(\mathbf{y}_0,\mathbf{Y})} = \{D_{i,0}, D_{i,1}\}_{i\in[1,n'+d+1]}$ and computes $\mathbf{t} = (\mathbf{t}[1], \ldots, \mathbf{t}[d])^\top \in \mathbb{Z}_p^d$ and $\mathbf{T} = (T_{i,j})_{(i,j)\in[1,d]\times[1,\hat{d}]} \in \mathbb{Z}_p^{d\times\hat{d}}$

such that $\hat{\mathbf{y}}_0 = \mathbf{y}_0 + \mathbf{Y}\mathbf{t}$ and $\hat{\mathbf{Y}} = \mathbf{Y}\mathbf{T}$. Such $\mathbf{t}$ and $\mathbf{T}$ exist because $\hat{\mathbf{y}}_0 + \mathrm{span}(\hat{\mathbf{Y}}) \subseteq \mathbf{y}_0 + \mathrm{span}(\mathbf{Y})$ and can be computed efficiently. Then it sets $\tilde{D}_{j,b}$ for $j \in [1, n' + \hat{d} + 1]$ and $b \in \{0, 1\}$ as

$$\tilde{D}_{j,b} = \begin{cases} D_{1,b} \cdot \prod_{i\in[1,d]} D_{1+i,b}^{\mathbf{t}[i]} & \text{if} \quad j = 1 \\ \prod_{i\in[1,d]} D_{1+i,b}^{T_{i,j-1}} & \text{if} \quad j \in [2, \hat{d}+1] \\ D_{j-\hat{d}+d,b} & \text{if} \quad j \in [\hat{d}+2, n'+\hat{d}+1] \end{cases}$$

Finally, it runs $\mathsf{ReRand}(\mathsf{mpk}, \{\tilde{D}_{i,0}, \tilde{D}_{i,1}\}_{i\in[1,n'+\hat{d}+1]}, (\hat{\mathbf{L}}, \hat{\rho})) \to \{D'_{i,0}, D'_{i,1}\}_{i\in[1,n'+\hat{d}+1]}$ and outputs $\mathsf{sk}'_{(\hat{\mathbf{y}}_0, \hat{\mathbf{Y}})} = \{D'_{i,0}, D'_{i,1}\}_{i\in[1,n'+\hat{d}+1]}$ where $(\hat{\mathbf{L}}, \hat{\rho}) = f_{\mathsf{k}}^{\mathsf{DSE}\to\mathsf{KP}}(\hat{\mathbf{y}}_0, \hat{\mathbf{Y}})$.

The following claim indicates that the key delegation algorithm described above correctly works.

Claim. For all $\lambda, n, (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}'(\lambda, n), \mathbf{y}_0, \mathbf{Y}, \hat{\mathbf{y}}_0, \hat{\mathbf{Y}}$ such that $\hat{\mathbf{y}}_0 + \mathrm{span}(\hat{\mathbf{Y}}) \subseteq \mathbf{y}_0 + \mathrm{span}(\mathbf{Y})$, and $\mathsf{sk}_{(\mathbf{y}_0, \mathbf{Y})} \leftarrow \mathsf{KeyGen}'(\mathsf{msk}, \mathsf{mpk}, (\mathbf{y}_0, \mathbf{Y}))$, we have the following distributions are identical:

$$\mathsf{KeyDel}'(\mathsf{mpk}, \mathsf{sk}_{(\mathbf{y}_0, \mathbf{Y})}, (\mathbf{y}_0, \mathbf{Y}), (\hat{\mathbf{y}}_0, \hat{\mathbf{Y}})) \approx \mathsf{KeyGen}'(\mathsf{mpk}, \mathsf{msk}, (\hat{\mathbf{y}}_0, \hat{\mathbf{Y}}))$$

Proof. Let $\{\tilde{D}_{i,0}, \tilde{D}_{i,1}\}_{i\in[1,n'+\hat{d}+1]}$ be defined as above which is computed from $\mathsf{sk}_{(\mathbf{L},\rho)} = \{D_{i,0} = f_0(\rho(i))^{r_i}, D_{i,1} = g^{\lambda_i} \cdot f_1(\rho(i))^{r_i}\}_{i\in[1,n'+d+1]}$ where $\lambda_i = \mathbf{L}_i \cdot \mathbf{s}$ for some $\mathbf{s} = (\alpha, \mathbf{s}[2], \ldots, \mathbf{s}[m'])^\top \in \mathbb{Z}_p^{m'}$ and $\{r_i \in \mathbb{Z}_p\}_{i\in[1,\ell]}$. Here, $m' = (n\kappa+1)(n+1)+1$. We will show that $\Pr[\{\tilde{D}_{i,0}, \tilde{D}_{i,1}\}_{i\in[1,n'+\hat{d}+1]} = \mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, (\hat{\mathbf{L}}, \hat{\rho}))] > 0$. Then, the output of KeyDel defined as above is correctly distributed due to Equation (7).

For $j \in [\hat{d}+2, n'+\hat{d}+1]$ and $b \in \{0, 1\}$, we have

$$\tilde{D}_{j,b} = g^{b\cdot\mathbf{L}_{j-\hat{d}+d}\cdot\mathbf{s}} \cdot f_b(\rho(j - \hat{d} + d))^{r_{j-\hat{d}+d}} = g^{b\cdot\hat{\mathbf{L}}_j\cdot\mathbf{s}} \cdot f_b(\hat{\rho}(j))^{r'_j}$$

where $r'_j = r_{j-\hat{d}+d}$ for $j \in [\hat{d}+2, n'+\hat{d}+1]$. The second equation above holds by the definition of $(\mathbf{L}, \rho)$ and $(\hat{\mathbf{L}}, \hat{\rho})$. For $j \in [2, \hat{d}+1]$ and $b \in \{0, 1\}$, we have that

$$\begin{aligned} \tilde{D}_{j,b} &= \prod_{i\in[1,d]} D_{1+i,b}^{T_{i,j-1}} = \prod_{i\in[1,d]} \left(g^{b\cdot\mathbf{L}_{1+i}\cdot\mathbf{s}} \cdot f_b(\rho(1+i))^{r_{1+i}}\right)^{T_{i,j-1}} \\ &= \left(g^{\sum_{i\in[1,d]} T_{i,j-1}\mathbf{L}_{1+i}\cdot\mathbf{s}}\right)^b \cdot f_b(\mathsf{D})^{\sum_{i\in[1,d]} r_{1+i}T_{i,j-1}} \\ &= \left(g^{\sum_{i\in[1,d]} T_{i,j-1}\mathbf{y}_i^\top\cdot\mathbf{s}'}\right)^b \cdot f_b(\mathsf{D})^{r'_j} \qquad \text{(By the structure of } \mathbf{L}.\text{)} \\ &= \left(g^{\hat{\mathbf{y}}_{j-1}^\top\cdot\mathbf{s}'}\right)^b \cdot f_b(\mathsf{D})^{r'_j} \qquad \text{(By the definition of } \mathbf{T}.\text{)} \\ &= \left(g^{\hat{\mathbf{L}}_j\cdot\mathbf{s}}\right)^b \cdot f_b(\hat{\rho}(j))^{r'_j} \qquad \text{(By the structure of } \hat{\mathbf{L}}.\text{)} \end{aligned}$$

In the above, $r'_j = \sum_{i\in[1,d]} r_{1+i}T_{i,j-1}$ for $j \in [2, \hat{d}+1]$, $\mathbf{s}' = (\mathbf{s}[3], \dots, \mathbf{s}[n+2])^\top$, and $\mathbf{y}_i$ is the i-th column of $\mathbf{Y}$. We also have

$$\begin{aligned}
\tilde{D}_{1,b} = D_{1,b} \cdot \prod_{i\in[1,d]} D_{1+i,b}^{\mathbf{t}[i]} &= g^{b\cdot\mathbf{L}_1\cdot\mathbf{s}} \cdot f_b(\rho(1))^{r_1} \cdot \prod_{i\in[1,d]} \left(g^{b\cdot\mathbf{L}_{1+i}\cdot\mathbf{s}} \cdot f_b(\rho(1+i))^{r_{1+i}}\right)^{\mathbf{t}[i]} \\
&= \left(g^{\left(\mathbf{L}_1 + \sum_{i\in[1,d]}\mathbf{t}[i]\mathbf{L}_{1+i}\right)\cdot\mathbf{s}}\right)^b \cdot f_b(\mathsf{D})^{r_1+\sum_{i\in[1,d]} r_{1+i}\mathbf{t}[i]} \\
&= \left(g^{\alpha+\mathbf{s}[2]+(\mathbf{y}_0^\top+\sum_{i\in[1,d]}\mathbf{t}[i]\mathbf{y}_i^\top)\cdot\mathbf{s}'}\right)^b \cdot f_b(\mathsf{D})^{r'_1} \qquad \text{(By the structure of } \mathbf{L}.) \\
&= \left(g^{\alpha+\mathbf{s}[2]+\hat{\mathbf{y}}_0^\top\cdot\mathbf{s}'}\right)^b \cdot f_b(\mathsf{D})^{r'_1} \qquad \text{(By the definition of } \mathbf{t}.) \\
&= \left(g^{\hat{\mathbf{L}}_1\cdot\mathbf{s}}\right)^b \cdot f_b(\hat{\rho}(1))^{r'_1} \qquad \text{(By the structure of } \hat{\mathbf{L}}.)
\end{aligned}$$

In the above, $r'_1 = r_1 + \sum_{i\in[1,d]} r_{1+i}\mathbf{t}[i]$.

To sum up, we have $(\tilde{D}_{i,0}, \tilde{D}_{i,1}) = (f_0(\hat{\rho}(i))^{r'_i}, g^{\hat{\mathbf{L}}_i\cdot\mathbf{s}} \cdot f_1(\hat{\rho}(i))^{r'_i})$ for $i \in [1, n'+\hat{d}+1]$. This implies that $(\tilde{D}_{i,0}, \tilde{D}_{i,1})$ can be seen as a private key for $(\hat{\mathbf{L}}, \hat{\rho})$ which is generated by randomness $\{r'_i\}_{i\in[n'+\hat{d}+1]}$ and $\{\mathbf{s}[i]\}_{i\in[2,m']}$. Thus we have

$$\Pr\left[\{\tilde{D}_{i,0}, \tilde{D}_{i,1}\}_{i\in[1,n'+\hat{d}+1]} = \mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, (\hat{\mathbf{L}}, \hat{\rho}))\right] > 0$$

as desired.

E From K(C)ASP to KP(CP)-ABE

In Section 5, we showed that monotonic KP(CP)-ABE can be converted into K(C)ASP. In this section, we also show the conversion direction. Namely, we show that KASP can be converted into non-monotonic KP-ABE with large universe (which trivially implies monotonic KP-ABE for small universe). The result in this section together with that in Section 5 imply that these two notions are in fact equivalent in the sense that one implies the other. These results also imply that monotonic KP-ABE for small universe and non-monotonic KP-ABE with large universe are equivalent. We note that we can obtain similar implications for the case of CP-ABE and CASP by swapping key and ciphertext attributes in the following.

E.1 The Conversion

We construct non-monotonic KP-ABE with large universe for $N = (\bar{k}, -, -, -)$ from KASP with dimension $\bar{k}+1$. As in Section 3, we define appropriate maps $f_{\mathsf{p}}^{\mathsf{KP}\to\mathsf{KASP}}$, $f_{\mathsf{e}}^{\mathsf{KP}\to\mathsf{KASP}}$, $f_{\mathsf{k}}^{\mathsf{KP}\to\mathsf{KASP}}$. At first, we define $f_{\mathsf{p}}^{\mathsf{KP}\to\mathsf{KASP}}(N) = \bar{k}+1$. We assume that the universe of attributes is $\mathbb{Z}_p$. This restriction can be easily removed by using collision resistant hash. The idea for the conversion is very similar to that in Section 4.

Next we define $f_{\mathsf{e}}^{\mathsf{KP}\to\mathsf{KASP}}$ that takes as input a set of attributes $S = (S_1, \dots, S_k) \subset \mathbb{Z}_p$ such that $k \le \bar{k}$ and outputs a vector $\mathbf{q}_S = (\mathbf{q}_S[1], \dots, \mathbf{q}_S[\bar{k}+1])^\top \in \mathbb{Z}_p^{\bar{k}+1}$ which is defined as a coefficient vector from

$$Q_S[Z] = \sum_{i=1}^{k+1} \mathbf{q}_S[i] \cdot Z^{i-1} = \prod_{i=1}^{k} (Z - S_i)$$

where, if $k < \bar{k}$, the coordinates $\mathbf{q}_S[k+2], \ldots, \mathbf{q}_S[\bar{k}+1]$ are all set to 0. We note that for $x \in \mathbb{Z}_p$, $Q_S(x) = \prod_{i=1}^{k}(x - S_i) = 0$ iff $x \in S$. This property will be used later.

Next we define $f_{\mathsf{k}}^{\mathsf{KP}\to\mathsf{KASP}}$ that takes as input a span program $(\mathbf{L} \in \mathbb{Z}_p^{\ell \times m}, \rho)$ and outputs an arithmetic span program $(\mathbf{Y}, \mathbf{Z}, \rho')$. Let the size of $\mathbf{L}$ be $(\ell_0 + \ell_1) \times m$ where $\ell = \ell_0 + \ell_1$ and let $\bar{\ell} = \max\{\ell_0, \ell_1\}$. Without loss of generality, we assume that the first ℓ_0 rows of the matrix are associated with positive attributes and the last ℓ_1 rows with negative attributes by a map ρ. We denote $\mathbf{L}$ as $\mathbf{L} = [\mathbf{L}_0; \mathbf{L}_1]$ using matrices $\mathbf{L}_0 \in \mathbb{Z}_p^{\ell_0 \times m}$ and $\mathbf{L}_1 \in \mathbb{Z}_p^{\ell_1 \times m}$. $\mathbf{Y}$ and $\mathbf{Z}$ are defined as

$$\mathbf{Y}^\top = \begin{pmatrix} & \mathbf{0}_{(\ell_0+2\ell_1)\times\left(2(\bar{k}+2)\ell+m\right)} & & & \\ \overbrace{}^{m} & \mathbf{I}_\ell \otimes \mathbf{1}_{\bar{k}+1} & \overbrace{}^{(\bar{k}+1)\bar{\ell}} & & \\ & & & \mathbf{I}_\ell \otimes \mathbf{1}_{\bar{k}+1} & \overbrace{}^{(\bar{k}+1)\bar{\ell}} \end{pmatrix} \in \mathbb{Z}_p^{\left(\ell_0+2\ell_1+2(\bar{k}+1)\bar{\ell}\right)\times\left(2(\bar{k}+2)\ell+m\right)}.$$

$$\mathbf{Z}^\top = \begin{pmatrix} \mathbf{L}_0 & \overbrace{}^{\bar{\ell}} & \mathbf{G}_0 & & & \\ \mathbf{L}_1 & & & \mathbf{I}_{\ell_1} & \overbrace{}^{\bar{\ell}-\ell_1} & \mathbf{G}_1 \\ & & \mathbf{I}_{(\bar{k}+1)\bar{\ell}} & & & \\ & & & & & \mathbf{I}_{(\bar{k}+1)\bar{\ell}} \end{pmatrix} \in \mathbb{Z}_p^{\left(\ell_0+2\ell_1+2(\bar{k}+1)\bar{\ell}\right)\times\left(2(\bar{k}+2)\ell+m\right)}.$$

Here, $\mathbf{G}_b \in \mathbb{Z}_p^{\ell_b \times \bar{\ell}(\bar{k}+1)}$ for $b \in \{0,1\}$ are defined as Equation (3). We also define $\rho' : [\ell_0 + 2\ell_1 + 2(\bar{k}+1)\bar{\ell}] \to [1, \bar{k}+1]$ as follows. If $1 \leq i \leq \ell_0 + 2\ell_1$, $\rho'(i) = 1$. If $i \geq \ell_0 + 2\ell_1 + 1$, we set $\rho'(i) = i'$ where $i' \in [1, \bar{k}+1]$ is a unique integer that satisfies $i - (\ell_0 + 2\ell_1) \equiv i' \mod (\bar{k}+1)$.

E.2 Correctness of the Conversion

By combining the following theorem with Lemma 1, we can see that monotonic KP-ABE with large universe can be constructed from KASP.

Theorem 4. *For any span program $(\mathbf{L}, \rho)$ and S such that $|S| \leq \bar{k}$, we have that*

$$R^{\mathsf{KASP}}(\mathbf{x}, (\mathbf{Y}, \mathbf{Z}, \rho')) = 1 \Leftrightarrow R^{\mathsf{KP}}(S, (\mathbf{L}, \rho)) = 1$$

where $\mathbf{x} = f_{\mathsf{e}}^{\mathsf{KP}\to\mathsf{KASP}}(S)$ and $(\mathbf{Y}, \mathbf{Z}, \rho') = f_{\mathsf{k}}^{\mathsf{KP}\to\mathsf{KASP}}(\mathbf{L}, \rho)$.

Proof. Let $\mathbf{W}$ be a matrix such that the i-th column is $\mathbf{x}[\rho'(i)] \cdot \mathbf{y}_i + \mathbf{z}_i$ where $\mathbf{y}_i$ and $\mathbf{z}_i$ are the i-th column of $\mathbf{Y}$ and $\mathbf{Z}$ respectively. Then, from the definition of ρ', we have that

$$\mathbf{W}^\top = \begin{pmatrix} \mathbf{L}_0 & & \mathbf{G}_0 & & & & \\ \mathbf{L}_1 & & & \mathbf{I}_{\ell_1} & \overbrace{}^{\bar{\ell}-\ell_1} & & \\ & & & & & \mathbf{G}_1 & \\ & \mathbf{H} & \mathbf{I}_{(\bar{k}+1)\bar{\ell}} & & & & \\ & & & \mathbf{H} & & & |\mathbf{I}_{(\bar{k}+1)\bar{\ell}} \end{pmatrix} \in \mathbb{Z}_p^{\left(\ell_0+2\ell_1+2(\bar{k}+1)\bar{\ell}\right)\times\left(2(\bar{k}+2)\ell+m\right)}$$

where $\mathbf{H} = \mathbf{I}_{\bar{\ell}} \otimes \mathbf{q}_S \in \mathbb{Z}_p^{(\bar{k}+1)\bar{\ell}\times\bar{\ell}}$. We divide the matrix $\mathbf{W}^\top$ into two parts as $\mathbf{W} = \left(\mathbf{W}_0, \mathbf{W}_1\right)$ where $\mathbf{W}_0 \in \mathbb{Z}_p^{\left(2(\bar{k}+2)\ell+m\right)\times\left(\ell_0+2\ell_1\right)}$ and $\mathbf{W}_1 \in \mathbb{Z}_p^{\left(2(\bar{k}+2)\ell+m\right)\times 2(\bar{k}+1)\bar{\ell}}$. Note that $\mathbf{W}_0$ and $\mathbf{W}_1$ defined here have exactly the same structure as $\mathbf{X}$ and $\mathbf{Y}$ defined in Section 4 respectively. We also let $I \subset [1, \ell(= \ell_0 + \ell_1)]$ be $I = \{i|\rho(i) \in S\}$. $\mathbf{L}_I$ is defined as the sub-matrix of $\mathbf{L}$ formed by rows whose index is in I. Then, we have

$$\begin{aligned} \mathbf{e}_1 \in \mathrm{span}(\mathbf{L}_I^\top) &\Leftrightarrow (-\mathbf{e}_1 + \mathrm{span}(\mathbf{W}_0)) \cap \mathrm{span}(\mathbf{W}_1) \neq \emptyset \\ &\qquad \text{(By exactly the same analysis as in the proof of the Claim in Section 4.2.)} \\ &\Leftrightarrow \mathbf{e}_1 \in \mathrm{span}(\mathbf{W}). \end{aligned}$$

By the definition of R^{KASP} and R^{KP}, this concludes the proof of the theorem.

F Details About Our Attribute-Based Signature Scheme

Here, we define predicate signature (PS) which is a signature analogue of predicate encryption. Attribute-based signature (ABS) can be captured as a special case of PS in which a private key is associated with a set of attributes while a signature is associated with a policy. In this section, we construct a PS scheme that is dual of ABS: a private key is associated with a policy and a signature with a set. By applying our KP-ABE-to-CP-ABE conversion technique developed in Section 3 and 4 to the PS scheme, we obtain the first ABS with constant signature size that supports non-monotone span program. Note that previous ABS scheme with constant-size signatures [24,10] only supports threshold predicate. Our result significantly improve the expressibility.

F.1 Predicate Signature

Here, we define predicate signature. Let $R = \{R_N : A_N \times B_N \to \{0,1\} \mid N \in \mathbb{N}^c\}$ be a relation family where A_N and B_N denote "verification attribute" and "key attribute" spaces and c is some fixed constant. The index $N = (n_1, n_2, \ldots, n_c)$ of R_N denotes the numbers of bounds for corresponding parameters. A predicate signature (PS) scheme Σ for R_N is defined by the following algorithms:

$\mathsf{Setup}(\lambda, N) \to (\mathsf{mpk}, \mathsf{msk})$: The setup algorithm is defined as the same as PE.

$\mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, Y) \to \mathsf{sk}_Y$: The key generation algorithm is defined as the same as PE.

$\mathsf{Sign}(\mathsf{mpk}, \mathsf{M}, X, \mathsf{sk}_Y, Y) \to \sigma$: The input to the signing algorithm is the master public key mpk, a message M, verification attribute X, private key sk_Y, and key attribute Y. It outputs a signature σ if $R_N(X, Y) = 1$ and otherwise $\perp$.

$\mathsf{Verify}(\mathsf{mpk}, \mathsf{M}, \sigma, X) \to 1$ or 0: The verification algorithm takes as input a master public key mpk, the message M, a signature σ, and a verification attribute $X \in A_N$. It outputs 1 if the signature is deemed valid and 0 otherwise. We assume that verification algorithm is deterministic.

We require correctness: that is, for all λ, N, $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(\lambda, N), X \in A_N, Y \in B_N$ such that $R(X, Y) = 1$, M in specified message space, $\mathsf{sk}_Y \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, Y)$, and $\sigma \leftarrow \mathsf{Sign}(\mathsf{mpk}, \mathsf{M}, X, \mathsf{sk}_Y, Y)$, $\mathsf{Verify}(\mathsf{mpk}, \mathsf{M}, \sigma, X) = 1$ holds.

For security of PS, we require privacy and unforgeability.

Privacy. We say that the PS scheme has complete privacy if for all λ, N, $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(\lambda, N)$, $X \in A_N$, $Y_0, Y_1 \in B_N$ satisfying $R_N(X, Y_0) = R_N(X, Y_1) = 1$, M in specified message space, and $\mathsf{sk}_{Y_b} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, Y_b)$ for $b \in \{0, 1\}$, the following distributions are the same.

$$\{\sigma_0 \leftarrow \mathsf{Sign}(\mathsf{mpk}, \mathsf{M}, X, \mathsf{sk}_{Y_0}, Y_0)\} \approx \{\sigma_1 \leftarrow \mathsf{Sign}(\mathsf{mpk}, \mathsf{M}, X, \mathsf{sk}_{Y_1}, Y_1)\}$$

Unforgeability. We now define the unforgeability for PS scheme Σ. This security notion is defined by the following game between a challenger and a forger $\mathcal{F}$.

At first, the challenger runs the setup algorithm and gives mpk to $\mathcal{F}$. Then $\mathcal{F}$ may adaptively make key-extraction queries. If $\mathcal{F}$ submits $Y \in B_N$ to the challenger, the challenger returns $\mathsf{sk}_Y \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, Y)$. $\mathcal{F}$ may also makes signing queries. If $\mathcal{F}$ submits (M, X, Y) such that $R_N(X, Y) = 1$ to the challenger, the challenger makes $\mathsf{sk}_Y \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, Y)$ and computes $\sigma \leftarrow \mathsf{Sign}(\mathsf{mpk}, \mathsf{M}, X, \mathsf{sk}_Y, Y)$ which is returned to $\mathcal{F}$. In the case of $R_N(X, Y) = 0$, the challenger returns $\perp$ to $\mathcal{F}$. At last, $\mathcal{F}$ outputs a forgery $(\mathsf{M}^\star, X^\star, \sigma^\star)$. We say that $\mathcal{F}$ succeeds if $\mathsf{Verify}(\mathsf{mpk}, \mathsf{M}^\star, \sigma^\star, X^\star) \to 1$, $\mathcal{F}$ never made a key-extraction query for Y such that $R(X^\star, Y) = 1$, and $\mathcal{F}$ never made a signing query for $(\mathsf{M}^\star, X^\star, Y)$ such that $R(X^\star, Y) = 1$. The advantage of a forger $Adv^{PS}_{\mathcal{F},\Sigma}$ is defined as the success probability of $\mathcal{F}$ in the above game. We say that Σ satisfies adaptive-predicate and message unforgeability under chosen message attack if $Adv^{PE}_{\mathcal{A},\Pi}$ is negligible for all probabilistic polynomial time (PPT) adversary $\mathcal{A}$.

A weaker notion called selective-predicate and adaptive-message unforgeability can be defined as in the above game with the exception that the adversary $\mathcal{A}$ has to choose $X^\star$ before seeing mpk but key-extraction queries, signing queries, and choice of $\mathsf{M}^\star$ can still be adaptive.

F.2 Linear Secret Sharing Scheme

We can construct secret sharing scheme for span program $(\mathbf{L}, \rho)$ as follows.

Definition 2 (Linear Secret Sharing Scheme). *A secret sharing scheme for span program $(\mathbf{L}, \rho)$ consists of the following two algorithms.*

$\mathsf{Share}_{\mathbf{L},\rho}$ *The algorithm takes as input $s \in \mathbb{Z}_p$ which is to be shared. Let $\mathbf{L}$ be an $\ell \times m$ matrix. It chooses $\mathbf{s}[2], \ldots, \mathbf{s}[m] \xleftarrow{\$} \mathbb{Z}_p$ and let $\mathbf{s} = (s, \mathbf{s}[2], \ldots, \mathbf{s}[m])^\top$. It outputs $\mathbf{L} \cdot \mathbf{s}$ as the vector of ℓ shares. The share $\lambda_i = \langle \mathbf{L}_i, \mathbf{s} \rangle$ belongs to party $\rho(i)$ for $i \in [1, \ell]$, where $\mathbf{L}_i$ denotes the i-th row of* $\mathbf{L}$.

$\mathsf{Recon}_{L,\pi}$ *The algorithm takes as input a set $S \subseteq \mathcal{U}$. Assume that the indicator vector δ of S is accepted by $(\mathbf{L}, \rho)$. Then it outputs a set of constants $\{(i, \mu_i)\}_{i\in I}$ which has a linear reconstruction property: $\sum_{i\in I} \mu_i \cdot \lambda_i = s$. Here, I is a set of indices whose labels of corresponding rows are set to 1 by the input δ.*

F.3 Embedding Lemma for PS

As noted in [23], a signature analogue of Lemma 1 holds.

Lemma 2. *Consider two relation families $R^{\mathsf{F}}_N : A_N \times B_N \to \{0,1\}$ and $R^{\mathsf{F}'}_{N'} : A'_{N'} \times B'_{N'} \to \{0,1\}$, which is parametrized by $N \in \mathbb{N}^c$ and $N' \in \mathbb{N}^{c'}$ respectively. Suppose that there exist efficient mappings $f_{\mathsf{p}} : \mathbb{Z}^{c'} \to \mathbb{Z}^c$, $f_{\mathsf{e}} : A'_{N'} \to A_{f_{\mathsf{p}}(N')}$, and $f_{\mathsf{k}} : B'_{N'} \to B_{f_{\mathsf{p}}(N')}$ such that f_{e} is injective and*

$$R^{\mathsf{F}'}_{N'}(X', Y') = 1 \Leftrightarrow R^{\mathsf{F}}_{f_{\mathsf{p}}(N')}(f_{\mathsf{e}}(X'), f_{\mathsf{k}}(Y')) = 1.$$

Then, a PS $\Sigma = \{\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}\}$ for R^{F}_N can be converted into a PS $\Sigma' = \{\mathsf{Setup}', \mathsf{KeyGen}', \mathsf{Sign}', \mathsf{Verify}'\}$ for $R^{\mathsf{F}'}_{N'}$ as $\mathsf{Setup}'(\lambda, N') = \mathsf{Setup}(\lambda, f_{\mathsf{p}}(N'))$, $\mathsf{KeyGen}'(\mathsf{msk}, \mathsf{mpk}, Y') = \mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, f_{\mathsf{k}}(Y'))$, $\mathsf{Sign}'(\mathsf{mpk}, \mathsf{M}, X', \mathsf{sk}_{Y'}, Y') = \mathsf{Sign}(\mathsf{mpk}, \mathsf{M}, f_{\mathsf{e}}(X'), \mathsf{sk}_{Y'}, f_{\mathsf{k}}(Y'))$, and $\mathsf{Verify}'(\mathsf{mpk}, \mathsf{M}, \sigma, X') = \mathsf{Verify}(\mathsf{mpk}, \mathsf{M}, \sigma, f_{\mathsf{e}}(X'))$. This conversion preserves security. Namely, if Σ is secure, so is Σ'. This holds for adaptive-predicate and message unforgeability, selective-predicate and adaptive-message unforgeability, and complete privacy.

Proof. Since we have $R^{\mathsf{F}'}_{N'}(X', Y'_0) = R^{\mathsf{F}'}_{N'}(X', Y'_1) = 1 \Leftrightarrow R^{\mathsf{F}}_{f_{\mathsf{p}}(N')}(f_{\mathsf{e}}(X'), f_{\mathsf{k}}(Y'_0)) = R^{\mathsf{F}}_{f_{\mathsf{p}}(N')}(f_{\mathsf{e}}(X'), f_{\mathsf{k}}(Y'_1)) = 1$ for any $X' \in A'_{N'}$ and $Y'_0, Y'_1 \in B'_{N'}$, complete privacy of Σ' is immediately followed by that of Σ. Next we show that if Σ is selective-attribute and adaptive-message unforgeable, so is Σ'. Other cases can be dealt similarly. We construct a forger $\mathcal{G}$ against Σ from a forger $\mathcal{F}$ against Σ'. At first, $\mathcal{F}$ submits its target $X'^{\star}$. Then, $\mathcal{G}$ submits $f_{\mathsf{e}}(X'^{\star})$ to its challenger. Then, $\mathsf{Setup}(\lambda, f_{\mathsf{p}}(N')) = \mathsf{Setup}'(\lambda, N') \to \mathsf{mpk}$ is run and mpk is given to $\mathcal{G}$. $\mathcal{G}$ gives mpk to $\mathcal{F}$. When $\mathcal{F}$ makes key query for $Y' \in B'_{N'}$, $\mathcal{G}$ computes $Y = f_{\mathsf{k}}(Y')$ and makes a key-extraction query for its challenger. Then, $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, f_{\mathsf{k}}(Y')) = \mathsf{KeyGen}'(\mathsf{msk}, \mathsf{mpk}, Y') \to \mathsf{sk}_{Y'}$ is run and $\mathcal{B}$ is given $\mathsf{sk}_{Y'}$. Then, $\mathcal{G}$ gives it to $\mathcal{F}$. Since we have $R^{\mathsf{F}}_{f_{\mathsf{p}}(N')}(f_{\mathsf{e}}(X'^{\star}), f_{\mathsf{k}}(Y')) = R^{\mathsf{F}'}_{N'}(X'^{\star}, Y') = 0$, the query is valid. When $\mathcal{F}$ makes signing query for (M, X', Y') such that $R_N(X', Y') = 1$, $\mathcal{G}$ submits $(\mathsf{M}, f_{\mathsf{e}}(X'), f_{\mathsf{k}}(Y'))$ to its challenger. Then, $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, f_{\mathsf{k}}(Y')) = \mathsf{KeyGen}'(\mathsf{msk}, \mathsf{mpk}, Y') \to \mathsf{sk}_{Y'}$ and $\mathsf{Sign}'(\mathsf{mpk}, \mathsf{M}, X', \mathsf{sk}_{Y'}, Y') = \mathsf{Sign}(\mathsf{mpk}, \mathsf{M}, f_{\mathsf{e}}(X'), \mathsf{sk}_{Y'}, f_{\mathsf{k}}(Y')) \to \sigma'$ are run. Then, σ' is returned to $\mathcal{G}$ and $\mathcal{G}$ gives it to $\mathcal{F}$. Finally, $\mathcal{F}$ outputs its forgery $(\mathsf{M}^{\star}, X'^{\star}, \sigma^{\star})$. Then, $\mathcal{G}$ outputs $(\mathsf{M}^{\star}, f_{\mathsf{e}}(X'^{\star}), \sigma^{\star})$ as its forgery. We claim that $\mathcal{G}$ succeeds whenever $\mathcal{F}$ succeeds. At first, we have $\mathsf{Verify}'(\mathsf{mpk}, \mathsf{M}^{\star}, \sigma^{\star}, X'^{\star}) = \mathsf{Verify}(\mathsf{mpk}, \mathsf{M}^{\star}, \sigma^{\star}, f_{\mathsf{e}}(X'^{\star}))$. Furthermore, all signing key queries made by $\mathcal{G}$ are of the form $(\mathsf{M}, f_{\mathsf{e}}(X'), f_{\mathsf{k}}(Y'))$ where (M, X', Y') is submitted by $\mathcal{F}$. We have that $\mathsf{M} = \mathsf{M}^{\star}$, $f_{\mathsf{e}}(X') = f_{\mathsf{e}}(X'^{\star})$, and $R^{\mathsf{F}}_{f_{\mathsf{p}}(N')}(f_{\mathsf{e}}(X'), f_{\mathsf{k}}(Y')) = 1$ if and only if $\mathsf{M} = \mathsf{M}^{\star}$, $X' = X'^{\star}$, and $R^{\mathsf{F}'}_{N'}(X'^{\star}, Y') = 1$. Here, we used the fact that f_{e} is an injective map. Therefore, $\mathcal{G}$ succeeds whenever $\mathcal{F}$. This concludes the proof.

F.4 The Construction

Here, we construct a PS scheme for relation R^{KP}_N where $N = (\bar{k}, -, -, -)$ with large universe. Namely, in the following scheme, the size of attribute set that is associated with a signature is bounded by $\bar{k}$, whereas other parameters are unbounded. Furthermore, all span program appearing in the scheme are monotone. For the sake of brevity, we assume that the message space of the following scheme to be $\{0,1\}^\kappa$. We also assume that the universe of attributes is $\mathcal{U} = \mathbb{Z}_p^* \backslash \mathcal{D}$ where $\mathcal{D} = \{\mathsf{D}_1, \ldots, \mathsf{D}_{\bar{k}}\} \subset \mathbb{Z}_p$ is a set of dummy attributes. These restrictions can be easily removed by using collision resistant hash and one can deal with message and attributes of any length.

$\mathsf{Setup}(\lambda, \bar{k})$: It samples $h_0, h_1, \ldots, h_{\bar{k}+1}, v_0, v_1, \ldots, v_\kappa \xleftarrow{\$} \mathbb{G}$. It also samples $\alpha \xleftarrow{\$} \mathbb{Z}_p$ and outputs master public key $\mathsf{mpk} = \{g, h_0, h_1, \ldots, h_{\bar{k}+1}, v_0, v_1, \ldots, v_\kappa, e(g,g)^\alpha\}$ and $\mathsf{msk} = \alpha$.

In the following, we denote $\mathsf{Wa}(\mathsf{M})$ for $v_0 \cdot \prod_{i \in \{j \in [\kappa] | \mathsf{M}[j]=1\}} v_i$ where $\mathsf{M}[j]$ is the j-th bit of M. Note that $\mathsf{Wa}(\mathsf{M})$ is efficiently computable from mpk.

$\mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, (\mathbf{L}, \rho))$: The input to the algorithm is the master secret key msk, the master public key mpk, and a monotone span program $(\mathbf{L}, \rho)$. Here, $\mathbf{L} \in \mathbb{Z}_p^{\ell \times m}$ and ρ is a map $\rho : [\ell] \to \mathcal{U}$. It chooses a vector $\mathbf{s} = (\mathbf{s}[1], \ldots, \mathbf{s}[m])^\top$ such that $\mathbf{s}[1] = \alpha$ and $\mathbf{s}[2], \ldots, \mathbf{s}[m] \xleftarrow{\$} \mathbb{Z}_p$ and calculates $\lambda_i = \mathbf{L}_i \cdot \mathbf{s}$ for each $i \in [\ell]$ where $\mathbf{L}_i$ is i-th *row* of $\mathbf{L}$. Finally, it chooses $r_i \xleftarrow{\$} \mathbb{Z}_p$ for all $i \in [\ell]$ and outputs private key

$$\mathsf{sk}_{(\mathbf{L},\rho)} = \left\{ \begin{array}{ll} D_{i,1} = g^{\lambda_i} \cdot h_0^{r_i}, & D_{i,2} = g^{r_i}, \\ \{K_{i,j} = (h_1^{-\rho(i)^j} h_{j+1})^{r_i}\}_{j \in [1,\bar{k}]}, & \{K'_{i,j} = v_j^{r_i}\}_{j \in [0,\kappa]} \end{array} \right\}_{i \in [\ell]} .$$

$\mathsf{Sign}(\mathsf{mpk}, \mathsf{M}, S, \mathsf{sk}_{(\mathbf{L},\rho)}, (\mathbf{L}, \rho))$: Assume that the monotone span program $(\mathbf{L}, \rho)$ associated to the private key $\mathsf{sk}_{(\mathbf{L},\rho)}$ satisfies $R^{\mathsf{KP}}(S, (\mathbf{L}, \rho)) = 1$ and otherwise it outputs 0. We let $I = \{i | \rho(i) \in S\}$. Then the signing algorithm can efficiently compute reconstruction coefficients $\{(i, \mu_i)\}_{i \in I} = \mathsf{Recon}_{\mathbf{L},\rho}(S)$ such that $\sum_{i \in I} \mu_i \lambda_i = \alpha$. It also sets $\hat{S} = S \cup \{\mathsf{D}_1, \ldots, \mathsf{D}_{\bar{k}-|S|}\}$. When $|S| = \bar{k}$, $\hat{S} = S$. Then it first computes

$$D'_{i,1} = D_{i,1} \cdot \prod_{j=1}^{\bar{k}} K_{i,j}^{\mathbf{q}_{\hat{S}}[j+1]} \cdot \prod_{j' \in \{0\} \cup \{j'' \in [1,\kappa] | \mathsf{M}[j'']=1\}} K'_{i,j'}$$

where $\mathbf{q}_{\hat{S}} = (\mathbf{q}_{\hat{S}}[1], \ldots, \mathbf{q}_{\hat{S}}[\bar{k}+1])^\top \in \mathbb{Z}_p^{\bar{k}+1}$ is defined as a coefficient vector from $Q_{\hat{S}}[Z] = \sum_{j=1}^{\bar{k}+1} \mathbf{q}_{\hat{S}}[j] \cdot Z^{j-1} = \prod_{\omega \in \hat{S}} (Z - \omega)$. Next it picks random $\tilde{r}$ and outputs the signature

$$\sigma = \left(\sigma_1 = \Big(\prod_{i \in I} (D'_{i,1})^{\mu_i}\Big) \cdot \Big(h_0 \prod_{j=1}^{\bar{k}+1} h_j^{\mathbf{q}_{\hat{S}}[j]}\Big)^{\tilde{r}} \cdot \mathsf{Wa}(\mathsf{M})^{\tilde{r}}, \quad \sigma_2 = \Big(\prod_{i \in I} D_{i,2}^{-\mu_i}\Big) \cdot g^{-\tilde{r}} \right) .$$

$\mathsf{Verify}(\mathsf{mpk}, \sigma, \mathsf{M}, S)$: To check that whether σ is a valid signature for a message M with set S, it first checks that $\sigma \in \mathbb{G}^2$. Otherwise it outputs 0. Then it checks

$$e(g,g)^{\alpha} \overset{?}{=} e(g,\sigma_1) \cdot e(h_0 \prod_{j=1}^{\bar{k}+1} h_j^{\mathbf{q}_{\hat{S}}[j]} \cdot \mathsf{Wa}(\mathsf{M}), \sigma_2)$$

and outputs 1 if it holds and 0 otherwise.

ABS with Constant-Size Signatures. The above scheme is PS scheme for relation R^{KP}. By combining the result in Section 3 and Lemma 2, we obtain a PS scheme for relation R^{DSE}. Furthermore, by combining the result in Section 4 and Lemma 2, we obtain the first non-monotonic ABS scheme with constant-size signatures. [5]

Here, we show the correctness and security of our PS scheme. Correctness of the scheme is implied by the following lemma and simple calculation. The lemma also implies privacy of the scheme since it indicates that the distribution of the signature only depends on M and S. In particular, it is completely independent from $\mathsf{sk}_{(\mathbf{L},\rho)}$.

Lemma 3. *For any λ, $\bar{k}$, κ, $\mathsf{M} \in \{0,1\}^{\kappa}$, $(\mathbf{L}, \rho)$, S, $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(\lambda, \bar{k})$, and a private key $\mathsf{sk}_{(\mathbf{L},\rho)} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, (\mathbf{L}, \rho))$ where S is accepted by $(\mathbf{L}, \rho)$, we have that σ generated by $\sigma \leftarrow \mathsf{Sign}(\mathsf{mpk}, \mathsf{M}, S, \mathsf{sk}_{(\mathbf{L},\rho)}, (\mathbf{L}, \rho))$ is distributed as follows:*

$$\sigma = \Big(\sigma_1 = g^{\alpha} \cdot \big(h_0 \prod_{j=1}^{\bar{k}+1} h_j^{\mathbf{q}_{\hat{S}}[j]} \cdot \mathsf{Wa}(\mathsf{M})\big)^{r}, \quad \sigma_2 = g^{-r}\Big)$$

where $r \xleftarrow{\$} \mathbb{Z}_p$.

Proof. We first observe that

$$\begin{aligned} D'_{i,1} &= g^{\lambda_i} \cdot h_0^{r_i} \cdot \big(\prod_{j \in [1,\bar{k}]} h_1^{-\rho(i)^j \mathbf{q}_{\hat{S}}[j+1]} h_{j+1}^{\mathbf{q}_{\hat{S}}[j+1]} \big)^{r_i} \cdot \mathsf{Wa}(\mathsf{M})^{r_i} \\ &= g^{\lambda_i} \cdot h_0^{r_i} \cdot h_1^{(-r_i)\cdot \left(\sum_{j\in[1,\bar{k}]} \rho(i)^j \mathbf{q}_{\hat{S}}[j+1]\right)} \cdot \big(\prod_{j \in [2,\bar{k}+1]} h_j^{\mathbf{q}_{\hat{S}}[j]} \big)^{r_i} \cdot \mathsf{Wa}(\mathsf{M})^{r_i} \\ &= g^{\lambda_i} \cdot \big(h_0 \cdot \prod_{j\in[1,\bar{k}+1]} h_j^{\mathbf{q}_{\hat{S}}[j]}\big)^{r_i} \cdot \mathsf{Wa}(\mathsf{M})^{r_i} \end{aligned}$$

where we used the fact that $-\sum_{j\in[1,\bar{k}]} \rho(i)^j \mathbf{q}_{\hat{S}}[j+1] = \mathbf{q}_{\hat{S}}[1]$ in the last equation. This holds since $i \in I$, we have $\rho(i) \in S \subseteq \hat{S}$ and $\sum_{j\in[1,\bar{k}+1]} \rho(i)^{j-1} \mathbf{q}_{\hat{S}}[j] = \prod_{\omega\in\hat{S}} (\rho(i) - \omega) = 0$.

[5] To apply Lemma 2, the map $f_{\mathsf{e}}^{\mathsf{CP}\to\mathsf{DSE}}$ should be injective. However, rigorously, it is not the case. This is because we adjust the number of columns of a matrix associated with a span program by padding zeroes in the computation of the map. This problem can be easily solved by restricting the size of matrices used in the ABS scheme to always be the same.

Next, we can see that

$$\begin{aligned}\sigma_1 &= \prod_{i \in I}\Big(g^{\lambda_i} \cdot \big(h_0 \cdot \prod_{j \in [1, \bar{k}+1]} h_j^{\mathbf{q}_{\hat{S}}[j]}\big)^{r_i} \cdot \mathsf{Wa}(\mathsf{M})\Big)^{\mu_i} \cdot \big(h_0 \cdot \prod_{j \in [1,\bar{k}+1]} h_j^{\mathbf{q}_{\hat{S}}[j]}\big)^{\tilde{r}} \cdot \mathsf{Wa}(\mathsf{M})^{\tilde{r}} \\ &= g^{\sum_{i \in I} \lambda_i \mu_i} \cdot \big(h_0 \cdot \prod_{j \in [1,\bar{k}+1]} h_j^{\mathbf{q}_{\hat{S}}[j]} \cdot \mathsf{Wa}(\mathsf{M})\big)^{\tilde{r} + \sum_{i \in I} r_i \mu_i} \\ &= g^{\alpha} \cdot \big(h_0 \prod_{j \in [1,\bar{k}+1]} h_j^{\mathbf{q}_{\hat{S}}[j]} \cdot \mathsf{Wa}(\mathsf{M})\big)^{r}\end{aligned}$$

and $\sigma_2 = \big(\prod_{i \in I} D_{i,2}^{-\mu_i}\big) \cdot g^{-\tilde{r}} = g^{-\tilde{r} - \sum_{i \in I} r_i \mu_i} = g^{-r}$ where $r = \tilde{r} + \sum_{i \in I} r_i \mu_i$. Here, r is uniformly distributed over $\mathbb{Z}_p$ due to $\tilde{r}$ as desired.

Unforgeability of the scheme is proven under the following assumption.

K-Computational Bilinear Diffie-Hellman Exponent (K-CBDHE) Assumption. We say that an adversary $\mathcal{A}$ breaks the K-CBDHE assumption on $(\mathbb{G}, \mathbb{G}_T)$ if

$$\Pr[\mathcal{A}(g, \{g^{a^i}\}_{i \in [2K] \setminus \{K+1\}}) \to g^{a^{K+1}}]$$

is non-negligible and $\mathcal{A}$ runs in polynomial time where $g \xleftarrow{\$} \mathbb{G}$, $a \xleftarrow{\$} \mathbb{Z}_p$.

Before going to the security proof, we introduce following lemma proved in [42] which abstracts out core technique used in the security proof of spatial encryption scheme proposed by Boneh and Hamburg [8].

Lemma 4. *([8]) Let $\mathbb{G}$ be a multiplicative group with prime order p and g be its generator. Let K, L be some integer bounded by polynomial of λ, $\mathbf{a}$ be $\mathbf{a} = (a, a^2, \ldots, a^K)^\top \in \mathbb{Z}_p^K$, $\{w_i\}_{i=0}^L$ be elements in $\mathbb{Z}_p$, and $\{\mathbf{u}_i\}_{i=0}^L$ be vectors in $\mathbb{Z}_p^K$. We also assume that $\mathbf{t} \in \mathbb{Z}_p^K$ satisfies $\langle \mathbf{t}, \mathbf{u}_0 \rangle \neq 0$ and $\langle \mathbf{t}, \mathbf{u}_i \rangle = 0$ for $i \in [1, L]$. Then, there exists an PPT* BHSim *which takes $(\{\mathbf{u}_i\}_{i=0}^L, \{w_i\}_{i=0}^L, \mathbf{t}, \{g^{a^i}\}_{i \in [1,2K] \setminus \{K+1\}})$ as input and outputs $(g^{a^{K+1}} \cdot (g^{\langle \mathbf{u}_0, \mathbf{a} \rangle + w_0})^r, \{(g^{\langle \mathbf{u}_i, \mathbf{a} \rangle + w_i})^r\}_{i=1}^L)$ where $r \xleftarrow{\$} \mathbb{Z}_p$.*

Theorem 5. *The above scheme is selective-predicate and adaptive-message unforgeable under chosen message attack if the $(\bar{k}+2)$-CBDHE assumption holds in $\mathbb{G}$.*

Proof. To prove the theorem, it suffices to show that there exists a PPT $\mathcal{A}$ that solves $(\bar{k}+2)$-CBDHE problem with non-negligible probability assuming a PPT forger $\mathcal{F}$ against our scheme with non-negligible advantage. We show this by considering the following sequences of the games. In the following, let X_i denote the probability that $\mathcal{F}$ is successful in Game i and the challenger does not abort. We also let $K = \bar{k} + 2$ for the simplicity of the notation.

Game **0.** We define Game 0 as an experiment between the challenger and the forger $\mathcal{F}$. Let the success probability of $\mathcal{F}$ in the game be $\Pr[X_0] = \epsilon$. By the assumption, we have ϵ is non-negligible.

Game **1.** In this game, we change the way to choose mpk and msk. At the outset of the game, $\mathcal{F}$ declares $S^\star$ which is the set she wants to be challenged upon. $\hat{S}^\star$ is defined as $\hat{S}^\star = S^\star \cup \{\mathsf{D}_1, \dots, \mathsf{D}_{\bar{k}-|S^\star|}\}$. Then, the challenger picks $a, \tilde{\alpha}, \tilde{h}_0, \dots, \tilde{h}_{\bar{k}+1}, \tilde{v}_0, \dots, \tilde{v}_\kappa, \xleftarrow{\$} \mathbb{Z}_p$, $\phi_0 \xleftarrow{\$} [-2\kappa Q, 2Q-1]$, $\phi_1, \dots, \phi_\kappa \xleftarrow{\$} [0, 2Q-1]$ and sets $\mathsf{msk} = a^{K+1} + \tilde{\alpha}$. Here, Q is the upper bound on the number of signing queries. Next, mpk is set as

$$\mathsf{mpk} = \begin{pmatrix} h_0 = g^{\tilde{h}_0} \cdot \prod_{i=1}^{\bar{k}+1} (g^{a^i})^{-\mathbf{q}_{\hat{S}^\star}[i]}, & h_i = g^{\tilde{h}_i} \cdot g^{a^i} \text{ for } i \in [1, \bar{k}+1], \\ e(g,g)^\alpha = e(g^a, g^{a^K}) \cdot e(g,g)^{\tilde{\alpha}}, & v_i = (g^{a^K})^{\phi_i} \cdot g^{\tilde{v}_i} \text{ for } i \in [0, \kappa] \end{pmatrix} \tag{8}$$

where $\mathbf{q}_{\hat{S}^\star}$ is defined as a coefficient vector from $Q_{\hat{S}^\star}[Z] = \sum_{j=1}^{\bar{k}+1} \mathbf{q}_{\hat{S}^\star}[j] \cdot Z^{j-1} = \prod_{\omega \in \hat{S}^\star} (Z - \omega)$. The rest of the game is the same as Game 0. Since the view of $\mathcal{F}$ is completely the same as that in Game 0, we have $\Pr[X_1] = \Pr[X_0]$.

Game **2.** Here, we define a function $\Phi : \{0,1\}^\kappa \to \mathbb{Z}_p$ and $\tilde{V} : \{0,1\}^\kappa \to \mathbb{Z}_p$ as

$$\Phi(\mathsf{M}) = \phi_0 + \sum_{i \in \{j \in [\kappa] | \mathsf{M}[j]=1\}} \phi_i, \qquad \tilde{V}(\mathsf{M}) = \tilde{v}_0 + \sum_{i \in \{j \in [\kappa] | \mathsf{M}[j]=1\}} \tilde{v}_i.$$

Note that we have $\mathsf{Wa}(\mathsf{M}) = (g^{a^K})^{\Phi(\mathsf{M})} \cdot g^{\tilde{V}(\mathsf{M})}$ by this definition. In this game, the challenger aborts if $\mathcal{F}$ makes signing query for $(\mathsf{M}, S^\star, (\mathbf{L}, \rho))$ such that $\Phi(\mathsf{M}) = 0$ and $R_N(S^\star, (\mathbf{L}, \rho)) = 1$. The challenger also aborts if the final output $(\mathsf{M}^\star, \sigma^\star)$ of $\mathcal{F}$ satisfies $\Phi(\mathsf{M}^\star) \neq 0$. By the same argument as the security proof of the Waters' identity-based encryption scheme [39], we have that $\Pr[X_2] \geq \Pr[X_1]/(32(\kappa+1)Q)$.

Finally, we replace the challenger in Game 2 with an algorithm $\mathcal{A}$ that solves CB-DHE problem with probability $\Pr[X_2]$ which is non-negligible assuming $\Pr[X_0]$ is non-negligible. We describe the description of $\mathcal{A}$ in the following.

Set-up of the Public Keys. Given the problem instance $(g, \{g^{a^i}\}_{i \in [1,2K] \setminus \{K+1\}})$ and $S^\star$ from $\mathcal{F}$, $\mathcal{A}$ picks $\tilde{\alpha}, \tilde{h}_0, \dots, \tilde{h}_{\bar{k}+1}, \tilde{v}_0, \dots, \tilde{v}_\kappa, \xleftarrow{\$} \mathbb{Z}_p$, $\phi_0 \xleftarrow{\$} [-2\kappa Q, 2Q-1]$, $\phi_1, \dots, \phi_\kappa \xleftarrow{\$} [0, 2Q-1]$, sets mpk as Equation (8), and gives it to $\mathcal{F}$. This step can be efficiently done only using problem instance, in particular, without knowing a or $g^{a^{K+1}}$.

Answering Private Key Queries. Throughout the game, $\mathcal{F}$ may ask for a private key for $(\mathbf{L}, \rho)$. Let $\mathbf{L}$ be an $\ell \times m$ matrix and I be $I = \{i \in [1, \ell] | \rho(i) \in \hat{S}^\star\}$. Since $\mathbf{L}$ does not accept $\hat{S}^\star$, we have $\mathbf{e}_1 \notin \mathrm{span}(\mathbf{L}_I^\top)$ where $\mathbf{L}_I$ is the sub-matrix of $(\mathbf{L}, \rho)$ formed by rows corresponding to index i such that $i \in I$. Hence, due to the proposition 11 in [22], we have that there must exists an efficiently computable vector $\mathbf{z} \in \mathbb{Z}_p^m$ such that $\langle \mathbf{e}_1, \mathbf{z} \rangle = 1$ and $\mathbf{L}_I \cdot \mathbf{z} = \mathbf{0}$. Now $\mathcal{A}$ picks $\mathbf{w}[2], \dots, \mathbf{w}[m] \xleftarrow{\$} \mathbb{Z}_p$ and implicitly defines $\mathbf{s} = (\mathbf{s}[1], \dots, \mathbf{s}[m])^\top = \alpha \mathbf{z} + \mathbf{w}$ where $\mathbf{w} = (0, \mathbf{w}[2], \dots, \mathbf{w}[m])^\top$. Note that we have that $\mathbf{s}[1] = \alpha$ and that $\mathbf{s}[2], \dots, \mathbf{s}[m] \in \mathbb{Z}_p$ are uniformly distributed, as required in Definition 2. Then $\mathcal{F}$ implicitly defines $\lambda_i = \mathbf{L}_i \cdot \mathbf{s}$ where $\mathbf{L}_i$ is the i-th row of $\mathbf{L}$. Then $\mathcal{A}$ computes $(D_{i,1}, D_{i,2}, \{K_{i,j}\}_{j \in [1, \bar{k}]}, \{K'_{i,j}\}_{j \in [0, \kappa]})$ for $i \in [1, \ell]$ as follows.

– For $i \in I$, we have that $\lambda_i = \mathbf{L}_i \cdot \mathbf{s} = \mathbf{L}_i \cdot \mathbf{w}$ holds since $\mathbf{L}_i \cdot \mathbf{z} = 0$ and thus λ_i can be efficiently computed by $\mathcal{A}$. In this case, $\mathcal{A}$ picks $r_i \xleftarrow{\$} \mathbb{Z}_p$ and computes $D_{i,1} = g^{\lambda_i} \cdot h_0^{r_i}$, $D_{i,2} = g^{r_i}$, $\{K_{i,j} = (h_1^{-\rho(i)^j} h_{j+1})^{r_i}\}_{j \in [1,\bar{k}]}$, and $\{K'_{i,j} = v_j^{r_i}\}_{j \in [0,\kappa]}$.

– For $i \notin I$, we have that $\lambda_i = \alpha \mathbf{L}_i \cdot \mathbf{z} + \mathbf{L}_i \cdot \mathbf{w} = \mu_1 a^{K+1} + \mu_2$ where $\mu_1 = \mathbf{L}_i \cdot \mathbf{z}$ and $\mu_2 = \mathbf{L}_i \cdot (\tilde{\alpha}\mathbf{z} + \mathbf{w})$. λ_i is known to $\mathcal{A}$ if $\mu_1 = 0$ and such a case can be dealt with as the same as the case of $i \in I$. Thus we assume that $\mu_1 \neq 0$ in the following. Since $\mathcal{A}$ does not know λ_i, we have to rely on Lemma 4 in order to answer the query.
$\mathcal{A}$ sets $\mathbf{u}_0 = (-\mathbf{q}_{\hat{S}^\star}^\top, 0)^\top \in \mathbb{Z}_p^K$, $w_0 = \tilde{h}_0$, $\mathbf{u}_1 = \mathbf{0}_K \in \mathbb{Z}_p^K$, $w_1 = 1$, $\mathbf{u}_{j+1} = -\rho(i)^j \mathbf{e}_1 + \mathbf{e}_{j+1} \in \mathbb{Z}_p^K$ and $w_{j+1} = -\rho(i)^j \tilde{h}_1 + \tilde{h}_{j+1}$ for $j \in [1, \bar{k}]$, and $\mathbf{u}_{j'+\bar{k}+2} = \phi_{j'} \mathbf{e}_K \in \mathbb{Z}_p^K$ and $w_{j'+\bar{k}+2} = \tilde{v}_{j'}$ for $j' \in [0, \kappa]$. It also sets $\mathbf{t} = (1, \rho(i), \ldots, \rho(i)^{K-1}, 0) \in \mathbb{Z}_p^K$. We can see that $\langle \mathbf{u}_0, \mathbf{t} \rangle = -\prod_{\omega \in \hat{S}^\star} (\rho(i) - \omega) \neq 0$ since $\rho(i) \notin \hat{S}^\star$ and $\langle \mathbf{u}_j, \mathbf{t} \rangle = 0$ for $j \in [1, \bar{k} + \kappa + 2]$.
Then $\mathcal{A}$ runs $\mathsf{BHSim}(\{\mathbf{u}_i\}_{i=0}^{\bar{k}+\kappa+2}, \{w_i\}_{i=0}^{\bar{k}+\kappa+2}, \mathbf{t}, \{g^{a^i}\}_{i \in [1,2K] \setminus \{K+1\}}) \to (g^{a^{K+1}} \cdot (g^{\langle \mathbf{u}_0, \mathbf{a} \rangle + w_0})^{\hat{r}}, \{(g^{\langle \mathbf{u}_i, \mathbf{a} \rangle + w_i})^{\hat{r}}\}_{i=1}^{\bar{k}+\kappa+2})$ where $\hat{r} \xleftarrow{\$} \mathbb{Z}_p$ due to Lemma 4. We have that $\hat{D}_{i,1} := g^{a^{K+1}} \cdot (g^{\langle \mathbf{u}_0, \mathbf{a} \rangle + w_0})^{\hat{r}} = g^{a^{K+1}} \cdot \big(g^{\tilde{h}_0} \cdot \prod_{i=1}^{\bar{k}+1} (g^{a^i})^{-\mathbf{q}_{\hat{S}^\star}[i]}\big)^{\hat{r}} = g^{a^{K+1}} \cdot h_0^{\hat{r}}$, $\hat{D}_{i,2} := (g^{\langle \mathbf{u}_1, \mathbf{a} \rangle + w_1})^{\hat{r}} = g^{\hat{r}}$, $\hat{K}_{i,j} := (g^{\langle \mathbf{u}_{j+1}, \mathbf{a} \rangle + w_{j+1}})^{\hat{r}} = \big((g^a)^{-\rho(i)^j} \cdot g^{a^{j+1}} \cdot g^{-\rho(i)^j \tilde{h}_1 + \tilde{h}_{j+1}}\big)^{\hat{r}} = (h_1^{-\rho(i)^j} h_{j+1})^{\hat{r}}$ for $j \in [1, \bar{k}]$, and $\hat{K}'_{i,j'} = (g^{\langle \mathbf{u}_{j'+\bar{k}+2}, \mathbf{a} \rangle + w_{j'+\bar{k}+2}})^{\hat{r}} = \big((g^{a^K})^{\phi_{j'}} \cdot g^{\tilde{v}_{j'}}\big)^{\hat{r}} = v_{j'}^{\hat{r}}$ for $j' \in [0, \kappa]$.
Finally, $\mathcal{A}$ sets $D_{i,1} = \hat{d}_{i,1}^{\mu_1} \cdot g^{\mu_2} = g^{\alpha} \cdot h_0^{r_i}$, $D_{i,2} = \hat{D}_{i,1}^{\mu_1} = g^{r_i}$, $K_{i,j} = \hat{K}_{i,j}^{\mu_1} = (h_1^{-\rho(i)^j} \cdot h_{j+1})^{r_i}$ for $j \in [1, \bar{k}]$, and $K'_{i,j'} = (\hat{K}'_{i,j'})^{\mu_i} = v_{j'}^{r_i}$ for $j' \in [0, \kappa]$ where $r_i = \mu_1 \hat{r}$. Since $\mu_1 \neq 0$ and $\hat{r}$ is uniformly distributed over $\mathbb{Z}_p$, r_i is also uniformly distributed over $\mathbb{Z}_p$ as desired.

Finally, $\mathcal{A}$ gives $\mathsf{sk}_{(\mathbf{L},\rho)} = \{D_{i,1}, D_{i,2}, \{K_{i,j}\}_{j \in [1,\bar{k}]}, \{K'_{i,j}\}_{j \in [0,\kappa]}\}_{i \in [1,\ell]}$ to $\mathcal{F}$.

Answering Signing Queries. When $\mathcal{F}$ makes a signing query to $(\mathsf{M}, S, (\mathbf{L}, \rho))$, $\mathcal{A}$ checks whether $R^{\mathsf{KP}}(S, (\mathbf{L}, \rho)) = 0$ and returns $\perp$ if it holds. Otherwise, it answers the signing query as follows. There are two cases to consider.

– In case of $S \neq S^\star$, we have that $\hat{S} \neq \hat{S}^\star$ where $\hat{S} = S \cup \{\mathsf{D}_1, \ldots, \mathsf{D}_{\bar{k}-|S|}\}$. Since $\hat{S} \neq \hat{S}^\star$ and $|\hat{S}| = |\hat{S}^\star|$, there exists $\mathsf{U} \in \mathcal{U} \cup \mathcal{D} = \mathbb{Z}_p^*$ such that $\mathsf{U} \notin \hat{S}^\star$ and $\mathsf{U} \in \hat{S}$. Then $\mathcal{A}$ sets a monotone span program $(\mathbf{L}', \rho')$ over $\mathcal{U} \cup \mathcal{D}$ such that $\mathbf{L}' = (1) \in \mathbb{Z}_p^{1 \times 1}$ and $\rho'(1) = \mathsf{U}$. The monotone span program accepts a set iff it includes U. So we have that $(\mathbf{L}', \rho')$ accepts $\hat{S}$ while it does not accept $\hat{S}^\star$. Then $\mathcal{A}$ can generates a private key $\mathsf{sk}_{(\mathbf{L}',\rho')} = \{g^{\alpha} \cdot h_0^r, g^r, \{(h_1^{-\mathsf{U}^j} h_{j+1})^r\}_{j \in [1,\bar{k}]}, \{v_{j'}^r\}_{j' \in [0,\kappa]}\}$ for $(\mathbf{L}', \rho')$ such that $r \xleftarrow{\$} \mathbb{Z}_p$ by the same way as answering private key queries. Note that such a monotone span program is not considered to be valid one in the real system when $\mathsf{U} \in \mathcal{D}$. We consider such a key only in the security proof. Then $\mathcal{A}$ runs $\sigma \leftarrow \mathsf{Sign}(\mathsf{mpk}, \mathsf{M}, S, \mathsf{sk}_{(\mathbf{L}',\rho')}, (\mathbf{L}', \rho'))$ and returns σ to $\mathcal{F}$. Due to the perfect privacy of the scheme, the distribution of σ is completely the same as honestly generated one.

– Next we consider the case of $S^\star = S$. First assume that $\mathit{\Phi}(\mathsf{M}) \neq 0$ since otherwise $\mathcal{A}$ aborts. $\mathcal{A}$ picks $\hat{r} \xleftarrow{\$} \mathbb{Z}_p$ and implicitly sets $r = -a/\mathit{\Phi}(\mathsf{M})+\hat{r}$. Then $\sigma = (\sigma_1, \sigma_2)$ is set as

$$\begin{aligned}\sigma_1 &= g^\alpha \cdot \big(h_0 \prod_{j\in[1,\bar{k}+1]} h_j^{\mathbf{q}_{\hat{S}^\star}[j]} \cdot \mathsf{Wa}(\mathsf{M})\big)^r \\ &= g^{a^{K+1}+\tilde{\alpha}} \cdot \big(g^{\tilde{h}_0+\sum_{i\in[1,\bar{k}+1]} \mathbf{q}_{\hat{S}^\star}[i]\tilde{h}_i} \cdot (g^{a^K})^{\mathit{\Phi}(\mathsf{M})} g^{\tilde{V}(\mathsf{M})}\big)^{-a/\mathit{\Phi}(\mathsf{M})+\hat{r}} \\ &= (g^a)^{-\tilde{h}} \cdot (g^{a^K})^{\mathit{\Phi}(\mathsf{M})\hat{r}} \cdot g^{\tilde{\alpha}+\hat{r}\tilde{h}} \qquad (9)\end{aligned}$$

and $\sigma_2 = (g^a)^{-1/\mathit{\Phi}(\mathsf{M})} \cdot g^{\hat{r}}$ where $\tilde{h} = \tilde{h}_0 + \sum_{i\in[1,\bar{k}+1]} \mathbf{q}_{\hat{S}^\star}[i]\tilde{h}_i + \tilde{V}(\mathsf{M})$. In the Equation (9), a problematic term $g^{a^{K+1}}$ cancels out. Thus $\mathcal{A}$ can efficiently compute σ_1 and σ_2 from the problem instance. Furthermore, since r is uniformly distributed over $\mathbb{Z}_p$, σ is correctly distributed by Lemma 3.

Extracting the Answer to the Problem from Forgery. At the last of the game, $\mathcal{F}$ outputs a forgery $(\mathsf{M}^\star, \sigma^\star = (\sigma_1^\star, \sigma_2^\star))$ for set $S^\star$. $\mathcal{A}$ aborts if $\mathit{\Phi}(\mathsf{M}^\star) \neq 0$ or $\mathsf{Verify}(\mathsf{mpk}, \sigma^\star, \mathsf{M}^\star, S^\star) = 0$. Otherwise, $\mathcal{A}$ can extract the answer to the CBDHE assumption as follows. Since $\mathsf{Verify}(\mathsf{mpk}, \sigma^\star, \mathsf{M}^\star, S^\star) = 1$, we have that $\sigma_2^\star = g^{-r}$ and

$$\begin{aligned}\sigma_1^\star &= g^\alpha \cdot \big(h_0 \prod_{j\in[1,\bar{k}+1]} h_j^{\mathbf{q}_{\hat{S}^\star}[j]} \cdot \mathsf{Wa}(\mathsf{M}^\star)\big)^r \\ &= g^{a^{K+1}+\tilde{\alpha}} \cdot (g^r)^{\tilde{h}_0+(\sum_{j\in[1,\bar{k}+1]} \mathbf{q}_{\hat{S}^\star}[j]\cdot\tilde{h}_j)+\tilde{V}(\mathsf{M}^\star)}\end{aligned}$$

for some $r \in \mathbb{Z}_p$. The second equation in the above follows from the fact that $\mathit{\Phi}(\mathsf{M}^\star) = 0$. Thus $\mathcal{A}$ can extract $g^{a^{K+1}}$ by computing $\sigma_1^\star \cdot g^{-\tilde{\alpha}} \cdot (\sigma_2^\star)^{\tilde{h}_0+(\sum_{j\in[1,\bar{k}+1]} \mathbf{q}_{\hat{S}^\star}[j]\cdot\tilde{h}_j)+\tilde{V}(\mathsf{M}^\star)} = g^{a^{K+1}}$.

Since the view of $\mathcal{F}$ is completely the same as that in Game 2, the success probability of $\mathcal{A}$ is $\Pr[X_2]$ which is non-negligible by the assumption.

Table of Contents