Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption

Allison Lewko *
University of Texas at Austin
alewko@cs.utexas.edu

Tatsuaki Okamoto
NTT
okamoto.tatsuaki@lab.ntt.co.jp

Amit Sahai †
UCLA
sahai@cs.ucla.edu

Katsuyuki Takashima
Mitsubishi Electric
Takashima.Katsuyuki@aj.MitsubishiElectric.co.jp

Brent Waters ‡
University of Texas at Austin
bwaters@cs.utexas.edu

Abstract

In this paper, we present two fully secure functional encryption schemes. Our first result is a fully secure attribute-based encryption (ABE) scheme. Previous constructions of ABE were only proven to be selectively secure. We achieve full security by adapting the dual system encryption methodology recently introduced by Waters and previously leveraged to obtain fully secure IBE and HIBE systems. The primary challenge in applying dual system encryption to ABE is the richer structure of keys and ciphertexts. In an IBE or HIBE system, keys and ciphertexts are both associated with the same type of simple object: identities. In an ABE system, keys and ciphertexts are associated with more complex objects: attributes and access formulas. We use a novel information-theoretic argument to adapt the dual system encryption methodology to the more complicated structure of ABE systems. We construct our system in composite order bilinear groups, where the order is a product of three primes. We prove the security of our system from three static assumptions. Our ABE scheme supports arbitrary monotone access formulas.

Our second result is a fully secure (attribute-hiding) predicate encryption (PE) scheme for inner-product predicates. As for ABE, previous constructions of such schemes were only proven to be selectively secure. Security is proven under a non-interactive assumption whose size does not depend on the number of queries. The scheme is comparably efficient to existing selectively secure schemes. We also present a fully secure hierarchical PE scheme under the same assumption. The key technique used to obtain these results is an elaborate combination of the dual system encryption methodology (adapted to the structure of inner product PE systems) and a new approach on bilinear pairings using the notion of dual pairing vector spaces (DPVS) proposed by Okamoto and Takashima.

*Supported by a National Defense Science and Engineering Graduate Fellowship.

†Research supported in part from NSF grants 0830803, 0627781, 0716389, 0456717, and 0205594, an equipment grant from Intel, and an Okawa Foundation Research Grant

‡Supported by NSF CNS-0716199, CNS-0915361, and CNS-0952692, Air Force Office of Scientific Research (AFO SR) under the MURI award for "Collaborative policies and assured information sharing" (Project PRE-SIDIO), Department of Homeland Security Grant 2006-CS-001-000001-02 (subaward 641), and the Alfred P. Sloan Foundation

1 Introduction

In a traditional public key encryption system, data is encrypted to be read by a particular individual who has already established a public key. Functional encryption is a new way of viewing encryption which opens up a much larger world of possibilities for sharing encrypted data. In a functional encryption system, there is a functionality $f(x, y)$ which determines what a user with secret key y can learn from a ciphertext encrypted under x (we can think of both x and y as binary strings, for example). This allows an encryptor to specify a policy describing what users can learn from the ciphertext, without needing to know the identities of these users or requiring them to have already set up public keys. The enhanced functionality and flexibility provided by such systems is very appealing for many practical applications.

Several previous works have pursued directions falling into this general framework, e.g. [36, 26, 18, 5, 34, 25, 41, 29, 13]. However, the same expressive power of these systems that makes them appealing also makes proving their security especially challenging. For this reason, all of the prior systems were only proven *selectively* secure, meaning that security was proven in a weaker model where part of the challenge ciphertext description must be revealed *before* the attacker receives the public parameters.

In this paper, we present fully secure systems for two cases of functional encryption, namely attribute-based encryption (ABE) and predicate encryption (PE) for inner products. Sahai and Waters [36] proposed Attribute-Based Encryption as a new concept of encryption algorithms that allow the encryptor to set a policy describing who should be able to read the data. In an attribute-based encryption system, private keys distributed by an authority are associated with sets of attributes and ciphertexts are associated with formulas over attributes. A user should be able to decrypt a ciphertext if and only if their private key attributes satisfy the formula. Predicate encryption for inner products was first presented by Katz, Sahai, and Waters [29]. In a predicate encryption scheme, secret keys are associated with predicates, and ciphertexts are associated with attributes. A user should be able to decrypt a ciphertext if and only if their private key predicate evaluates to 1 when applied to the ciphertext attribute.

Our Two Results The ABE and PE schemes described in this paper have essential commonalities: both are functional encryption schemes that employ the dual system methodology of Waters [42] to prove full security. This is a powerful tool for achieving full security of systems with advanced functionalities, but realizing the dual system methodology in each new context presents unique challenges. In particular, the technical challenges for ABE and PE are distinct, and the two results now combined into this paper were obtained by separate research groups working independently. The ABE result was obtained by Lewko, Sahai, and Waters, while the PE result was obtained by Okamoto and Takashima.

1.1 Attribute-Based Encryption

We are particularly interested in attribute-based encryption as a special case of functional encryption because it provides a functionality that can be very useful in practice. For example, a police force could use an ABE system to encrypt documents under policies like "Internal Affairs OR (Undercover AND Central)" and give out secret keys to undercover officers in the central division corresponding to the attributes "Undercover" and "Central". Given the many potential uses of ABE systems, constructing efficient systems with strong security guarantees is an important problem.

Previous Constructions and Selective Security. All previous constructions of ABE systems [36, 26, 19, 5, 34, 25, 41] have only been proven to be selectively secure. This is a limited model of security where the attacker is required to announce the target he intends to attack before seeing the public parameters of the system. This is an unnatural and undesirable restriction on the attacker, but it unfortunately appears to be necessary for the proof techniques used in prior works.

To see why this is the case, it is instructive to look into the way that previous security proofs have worked. In these security proofs, the simulator uses the attacker's announced target to embed the challenge in the public parameters in such a way that the simulator can produce any keys the attacker can request but can also leverage the attacker's output to break the underlying challenge. This is a *partitioning* strategy reminiscent of the strategies first used to prove security for IBE systems. The formation of the public parameters partitions the keys into two classes: those that the simulator can make, and those that are useful to the simulator in solving its challenge.

While this partitioning strategy was successfully employed by Boneh and Boyen [7], and Waters [40] to prove full security for an IBE system, any partitioning approach seems doomed to failure when one tries to achieve full security for ABE systems. Without selectivity, the simulator cannot anticipate which keys the attacker may ask for, so the attacker must make some type of a guess about what the partition should be. One natural direction is to partition the identity space in some random way and hope that the attacker's queries respect the partition (which was the main idea behind the works in the IBE setting). For ABE systems, however, private keys and ciphertexts have much more structure; different keys can be related (they may share attributes), and this severely restricts allowable partitions. Thus, the power and expressiveness of ABE systems work directly against us when attempting to create partitioning proofs.

Our Approach. We are able to obtain full security by adapting the dual system encryption technique of [42, 30] to the ABE case. Waters [42] introduced dual system encryption to overcome the limitations of partitioning. In a dual encryption system, keys and ciphertexts can take on one of two forms: normal and semi-functional. A normal key can decrypt both normal and semi-functional ciphertexts, while a semi-functional key can only decrypt normal ciphertexts. The semi-functional keys and ciphertexts are not used in the real system, only in the proof of security. The proof employs a hybrid argument over a sequence of security games. The first is the real security game, with normal keys and ciphertext. In the second game, the ciphertext is semi-functional and the keys remain normal. In subsequent games, the keys requested by the attacker are changed to be semi-functional one by one. By the final game, none of the keys given out are actually useful for decrypting a semi-functional ciphertext, and proving security becomes relatively easy.

There is one important subtlety inherent in the dual system technique. In the step where the k^{th} key becomes semi-functional, the simulator must be prepared to make any semi-functional challenge ciphertext and any key as the k^{th} key. At first, this appears to be a paradox, since it seems the simulator can just make a key that should decrypt the challenge ciphertext and decide for itself whether the key is semi-functional by attempting to decrypt the semi-functional challenge ciphertext. Waters addresses this issue by introducing tags: if a key and ciphertext in his IBE system have the same tag, decryption will fail *regardless* of semi-functionality. The simulator is constructed in such a way that if it attempts to check if key k is semi-functional by decrypting a semi-functional ciphertext, it will be thwarted because they will have equal tags. (This relationship between the tags will be hidden to an attacker who cannot request a key able to decrypt the challenge ciphertext.)

Lewko and Waters [30] provide a new realization of dual system encryption where tags are replaced by *nominally* semi-functional keys. Nominally semi-functional keys are structured like semi-functional keys except that they do also successfully decrypt semi-functional ciphertexts (the semi-functional contribution cancels out). When the k^{th} key turns semi-functional in the hybrid, the simulator is constructed so that it can only make a *nominally* semi-functional key k. It is then argued that this looks like a regular semi-functional key to the attacker.

Though they achieve fully secure HIBE with constant size ciphertext, it is not clear how to extend the techniques of [42, 30] to obtain fully secure ABE systems. Both rely on the fact that the identities attached to keys and ciphertexts are the same. Waters relies on this to align tags, while Lewko and Waters use this symmetry in designing their system so that a nominally semi-functional key is identically distributed to a regular semi-functional key in the view of an attacker who cannot decrypt. This symmetry does not hold in an ABE system, where keys and ciphertexts are each associated with different objects: attributes and formulas. The additional flexibility and expressiveness of ABE systems leads to a much more complicated structure of relationships between keys and ciphertexts, which makes the potential paradox of the dual system encryption technique more challenging to address for ABE.

We overcome this by giving a new realization of *nominally* semi-functional keys in the ABE setting. We do this by designing the semi-functional components of our keys and ciphertexts to mirror the functionality of the ABE scheme. Intuitively, we want to argue that an attacker who cannot decrypt the message also cannot determine if the final contribution of the semi-functional components will be non-zero. We make this argument information-theoretically by showing that our nominally semi-functional keys are distributed identically to regular semi-functional keys from the attacker's perspective. This information-theoretic argument is more intricate than the HIBE analog executed in [30], due to the more complicated structure of ABE systems.

The ideas above allow us to construct an ABE system that is fully secure. We build our construction in two phases. First, we construct an ABE system with the restriction that each attribute can only be used once in an access formula. We call this a *one-use* ABE system. Then, we provide a generic transformation from a one-use system to a system which is fully secure when attributes are used multiple times (up to a constant number of uses fixed at setup). While this transformation does incur some cost in key size, it does not increase the size of the ciphertext; we stress that ours is the first feasibility result for fully secure ABE. Our construction supports arbitrary monotone access formulas. We realize our ABE construction using bilinear groups of composite order and prove security under three assumptions used by Lewko and Waters [30].

1.2 Predicate Encryption for Inner Products

ABE systems have desirable functionality, but have one limitation in that the structure of the ciphertext is revealed to users who cannot reveal. For example, in a CP-ABE system, a user who cannot decrypt can still learn the formula associated with the ciphertext. For applications where the access policy must also be kept secret, this is unacceptable. In our second result we address a class of systems, called predicate encryption systems, that overcome this limitation. Our second result gives predicate encryption of inner products between the ciphertext and key vectors.

Predicate encryption (PE) for inner products was presented by Katz, Sahai and Waters [29] as a generalized (fine-grained) notion of encryption that covers identity-based encryption (IBE) [6, 7, 9, 20, 22, 28], hidden-vector encryption (HVE) [13] and attribute-based encryption (ABE) [5, 26, 34, 35, 36]. Informally, secret keys in a PE scheme correspond to *predicates* in some class $\mathcal{F}$, and a sender associates a ciphertext with an *attribute* in set Σ; a ciphertext associated with attribute $I \in \Sigma$ can be decrypted using a secret key sk_f corresponding to predicate $f \in \mathcal{F}$ if

and only if $f(I) = 1$.

The special case of inner product predicates is obtained by having each attribute correspond to a vector $\overrightarrow{x}$ and each predicate $f_{\overrightarrow{v}}$ correspond to a vector $\overrightarrow{v}$, where $f_{\overrightarrow{v}}(\overrightarrow{x}) = 1$ iff $\overrightarrow{x} \cdot \overrightarrow{v} = 0$. (Here, $\overrightarrow{x} \cdot \overrightarrow{v}$ denotes the standard inner-product). We note that these represent a wide class of predicates including equality tests (for IBE and HVE), disjunctions or conjunctions of equality tests, and, more generally, arbitrary CNF or DNF formulas (for ABE). However, we note that inner product predicates are less expressive than the LSSS access structures of ABE. To use inner product predicates for ABE, formulas must be written in CNF or DNF form, which can cause a superpolynomial blowup in size for arbitrary formulas.

Katz, Sahai, and Waters also introduced *attribute-hiding*, a security notion for PE that is stronger than the basic security requirement, *payload-hiding*. Roughly speaking, attribute-hiding requires that a ciphertext conceal the associated attribute as well as the plaintext, while payload-hiding only requires that a ciphertext conceal the plaintext. If attributes are identities, i.e., PE is IBE, attribute-hiding PE implies *anonymous* IBE. This notion of attribute-hiding addresses the limitation of ABE systems. Katz, Sahai, and Waters provided a scheme which is attribute-hiding PE for inner-product predicates, but it is only proven to be selectively secure and no delegation functionality is provided.

Our Results

- This paper proposes the first *adaptively secure* PE scheme for *inner-product* predicates in the *standard model*. The scheme is proven to be adaptively attribute-hiding (against CPA) under an assumption that is *non-interactive*. The number of terms of the assumption depends on a system parameter n, which is the vector length. (However, the number of terms does not depend on the number of adversarial private key queries.) We prove that the assumption is true in the generic model of bilinear pairing groups.

  The efficiency of the proposed PE scheme is comparable to that of the existing *selectively-secure* PE schemes [29, 33].

- This paper also establishes a (hierarchical) delegation functionality on the proposed adaptively secure PE scheme. That is, we propose an *adaptively secure* (attribute-hiding) hierarchical PE (HPE) scheme for *inner-product* predicates (with polynomially many levels) in the *standard model* under the n-eDDH assumption.

  The proposed HPE scheme implies the first *anonymous* hierarchical IBE (HIBE) with polynomially many levels in the standard model as a special case (when the associated inner-product predicate is specialized as the equality test for HIBE).

- It is straightforward to convert the (CPA-secure) basic (H)PE scheme to a CCA-secure (H)PE scheme by employing an existing general conversion such as that by Canetti, Halevi and Katz [17] or that by Boneh and Katz [12] (using an additional level with two-dimensions for the basic (H)PE scheme, and a strongly unforgeable one-time signature scheme or message authentication code and encapsulation). That is, we can present a *fully secure* (adaptively attribute-hiding against CCA) (H)PE scheme for *inner-product* predicates in the *standard model* under the n-eDDH assumption as well as a strongly unforgeable one-time signature scheme or message authentication code and encapsulation.

- To achieve the result, this paper elaborately combines a new methodology, the dual system encryption, proposed by Waters [42] and a new approach based on a notion of higher dimensional vector spaces, *dual pairing vector spaces* (DPVS), proposed by Okamoto and Takashima [32, 33]. The notion of DPVS is constructed on bilinear pairing groups, and

they presented a selectively secure (H)PE scheme on DPVS [33]. We will explain this approach and our key technique in Section 3.1.

Note that the n-eDDH assumption in this paper is defined over the basic primitive, bilinear pairing groups (not over the higher level concept, DPVS), although the proposed PE and HPE schemes are constructed over DPVS, and the assumptions in [33] are defined over DPVS.

- Since HPE is a generalized (fine-grained) version of anonymous HIBE (AHIBE) (or includes AHIBE as a special case), HPE covers (a generalized version of) applications described in [14], fully private communication and search on encrypted data. For example, we can use a two-level HPE scheme where the first level corresponds to the predicate/attribute of (single-layer) PE and the second level corresponds to those of "attribute search by a predicate" (generalized "key-word search").

1.3 Related Work

Identity Based Encryption (IBE) was proposed by Shamir [37]. In an identity based encryption system, an authority distributes keys to users with associated identities, and messages are encrypted directly to identities. The first IBE schemes were constructed by Boneh and Franklin [9] and Cocks [20]. These schemes were proven secure in the random oracle model. Then selectively secure schemes in the standard model were constructed [16, 6]. Boneh and Boyen [7] and Waters [40] constructed fully secure IBE schemes in the standard model. Gentry [22] gave an IBE system and security proof that moved beyond the confines of the partitioning strategy, but at the cost of a large and complicated complexity assumption.

Hierarchical Identity Based Encryption (HIBE) [24, 28] expands the functionality of identity based encryption to include a hierarchical structure on identities, where identities can delegate secret keys to their subordinate identities. Boneh and Boyen [6] constructed a selectively secure HIBE scheme. Boneh, Boyen, and Goh [8] constructed a selectively secure HIBE scheme with constant size ciphertexts. Gentry and Halevi [23] extended Gentry's techniques to get a fully secure HIBE system, but under "q-type" assumptions. Waters [42] leveraged the dual system encryption methodology to obtain fully secure IBE and HIBE systems from simple assumptions. Lewko and Waters [30] extended the dual encryption technique to obtain a fully secure HIBE system with constant size ciphertexts.

Attribute-based encryption was introduced by Sahai and Waters [36]. Goyal, Pandey, Sahai, and Waters [26] formulated two complimentary forms of ABE: Ciphertext-Policy Attribute-Based Encryption (CP-ABE) and Key-Policy Attribute-Based Encryption (KP-ABE). In a CP-ABE system, keys are associated with sets of attributes and ciphertexts are associated with access policies. In a KP-ABE system, the situation is reversed: keys are associated with access policies and ciphertexts are associated with sets of attributes. Selectively secure CP-ABE and KP-ABE systems were constructed in [36, 26, 19, 5, 34, 25, 41].

Goyal, Jain, Pandey, and Sahai [25] provide a general way to transform a KP-ABE system into a CP-ABE system. Chase [18] considered the problem of ABE with multiple authorities.

Other works have discussed similar problems without addressing collusion resistance [1, 2, 3, 15, 31, 39]. In these systems, the data encryptor specifies an access policy such that a set of users can decrypt the data only if the *union* of their credentials satisfies the access policy.

Predicate encryption was introduced by Katz, Sahai, and Waters [29], who also provided a scheme which is attribute-hiding PE for inner-product predicates; only the selective security (not adaptive security) is proven and no delegation functionality is provided.

Shi and Waters [38] presented a delegation mechanism for a class of PE, but the admissible

predicates of the system, which is a class of equality tests for HVE, are more limited than inner-product predicates in [29]. Moreover, they proved only selective security.

Okamoto and Takashima [33] proposed a (hierarchical) delegation mechanism for a PE scheme, i.e., a hierarchical PE (HPE) scheme, for inner-product predicates, but only selective security is proven.

Dual pairing vector spaces were introduced by Okamoto and Takashima [32, 33], who presented a selectively secure (H)PE scheme based on DPVS.

1.4 Organization

In Section 2, we present our result for ABE. In more detail, Subsection 2.1 provides the necessary background on access structures, linear secret-sharing schemes, CP-ABE, composite order bilinear groups, and states our complexity assumptions. Subsection 2.2, we describe our transformation from a one-use CP-ABE system to a system that is secure when attributes are used multiple times in a formula. In Subsection 2.3, we present our CP-ABE system and prove its security. In Subsection 2.4, we discuss extensions of our ABE result.

In Section 3, we present our result for PE for inner products. Subsection 3.1 describes the main ideas of the approach and establishes the necessary notations. In Subsection 3.2, we formally define DPVS. In Subsection 3.3, we state the complexity assumption. In Subsection 3.4, we formally define predicate encryption and inner product predicate encryption. In Subsection 3.5, we present our inner product predicate encryption scheme and its security. In Subsection 3.6, we present our HPE scheme.

2 Fully Secure Attribute-Based Encryption

2.1 Background

We first define access structures and linear secret-sharing schemes (LSSS). We then formally define CP-ABE and give the full security definition. We also give the necessary background on composite order bilinear groups and state our complexity assumptions.

2.1.1 Access Structures

Definition 1 *(Access Structure [4]) Let $\{P_1, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1,\ldots,P_n\}}$ is monotone if $\forall B, C$: if $B \in \mathbb{A}$ and $B \subseteq C$, then $C \in \mathbb{A}$. An* access structure *(respectively, monotone access structure) is a collection (respectively, monotone collection) $\mathbb{A}$ of non-empty subsets of $\{P_1, \ldots, P_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{P_1,\ldots,P_n\}} \setminus \{\}$. The sets in $\mathbb{A}$ are called the* authorized sets*, and the sets not in $\mathbb{A}$ are called the* unauthorized sets*.*

In our setting, attributes will play the role of parties and we will only deal with monotone access structures. We note that it is possible to (inefficiently) realize general access structures with our techniques by having the negation of an attribute be a separate attribute (so the total number of attributes will be doubled).

Linear Secret-Sharing Schemes Our construction will employ linear secret-sharing schemes (LSSS). We use the definition adapted from [4].

Definition 2 *(Linear Secret-Sharing Schemes (LSSS)) A secret sharing scheme Π over a set of parties $\mathcal{P}$ is called* linear *(over $\mathbb{Z}_p$) if*

1. *The shares for each party form a vector over $\mathbb{Z}_p$.*

2. *There exists a matrix A called the share-generating matrix for Π. The matrix A has ℓ rows and n columns. For all $i = 1, \ldots, \ell$, the i^{th} row of A is labeled by a party $\rho(i)$ (ρ is a function from $\{1, \ldots, \ell\}$ to $\mathcal{P}$). When we consider the column vector $v = (s, r_2, \ldots, r_n)$, where $s \in \mathbb{Z}_p$ is the secret to be shared and $r_2, \ldots, r_n \in \mathbb{Z}_p$ are randomly chosen, then Av is the vector of ℓ shares of the secret s according to Π. The share $(Av)_i$ belongs to party $\rho(i)$.*

We note the *linear reconstruction* property: we suppose that Π is an LSSS for access structure $\mathbb{A}$. We let S denote an authorized set, and define $I \subseteq \{1, \ldots, \ell\}$ as $I = \{i | \rho(i) \in S\}$. Then the vector $(1, 0, \ldots, 0)$ is in the span of rows of A indexed by I, and there exist constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that, for any valid shares $\{\lambda\}_i$ of a secret s according to Π, we have: $\sum_{i \in I} \omega_i \lambda_i = s$. These constants $\{\omega_i\}$ can be found in time polynomial in the size of the share-generating matrix A [4]. We note that for unauthorized sets, no such constants $\{\omega_i\}$ exist.

For our composite order group construction, we will employ LSSS matrices over $\mathbb{Z}_N$, where $N = p_1 p_2 p_3$ is a product of three distinct primes. As in the definition above over $\mathbb{Z}_p$, we say a set of attributes S is authorized if the rows of the access matrix A labeled by attributes in S have the vector $(1, 0, \ldots, 0)$ in their span modulo N. However, in our security proof for our composite order system, we will further assume that for an unauthorized set, the corresponding rows of A do not include the vector $(1, 0, \ldots, 0)$ in their span modulo p_2. We may assume this because if an adversary can produce an access matrix A over $\mathbb{Z}_N$ and an unauthorized set over $\mathbb{Z}_N$ that is authorized over $\mathbb{Z}_{p_2}$, then this can be used to produce a non-trivial factor of the group order N, which can be used to break our security assumptions.

Boolean Formulas Access policies might also be described in terms of monotonic boolean formulas. LSSS access structures are more general and can be derived from such representations. More precisely, one can use standard techniques to convert any monotonic boolean formula into a corresponding LSSS matrix. We can represent the boolean formula as an access tree, where the interior nodes are AND and OR gates, and the leaf nodes correspond to attributes. The number of rows in the corresponding LSSS matrix will be same as the number of leaf nodes in the access tree.

2.1.2 CP-ABE

A ciphertext-policy attribute-based encryption system consists of four algorithms: Setup, Encrypt, KeyGen, and Decrypt.

Setup$(\lambda, U) \rightarrow (PK, MSK)$ The setup algorithm takes in the security parameter λ and the attribute universe description U. It outputs the public parameters PK and a master secret key MSK.

Encrypt$(PK, M, \mathbb{A}) \rightarrow CT$ The encryption algorithm takes in the public parameters PK, the message M, and an access structure $\mathbb{A}$ over the universe of attributes. It will output a ciphertext CT such that only users whose private keys satisfy the access structure $\mathbb{A}$ should be able to extract M. We assume that $\mathbb{A}$ is implicitly included in CT.

KeyGen$(MSK, PK, S) \rightarrow SK$ The key generation algorithm takes in the master secret key MSK, the public parameters PK, and a set of attributes S. It outputs a private key SK.

Decrypt$(PK, CT, SK) \to M$ The decryption algorithm takes in the public parameters PK, a ciphertext CT, and a private key SK. If the set of attributes of the private key satisfies the access structure of the ciphertext, it outputs the message M.

2.1.3 Security Model for CP-ABE

We now give the full security definition for CP-ABE systems. This is described by a security game between a challenger and an attacker. The game proceeds as follows:

Setup The challenger runs the Setup algorithm and gives the public parameters PK to the attacker.

Phase 1 The attacker queries the challenger for private keys corresponding to sets of attributes $S_1, \ldots, S_{q_1}$.

Challenge The attacker declares two equal length messages M_0 and M_1 and an access structure $\mathbb{A}^*$. This access structure cannot be satisfied by any of the queried attribute sets $S_1, \ldots, S_{q_1}$. The challenger flips a random coin $\beta \in \{0, 1\}$, and encrypts M_b under $\mathbb{A}^*$, producing CT^*. It gives CT^* to the attacker.

Phase 2 The attacker queries the challenger for private keys corresponding to sets of attributes $S_{q_1+1}, \ldots, S_q$, with the added restriction that none of these satisfy $\mathbb{A}^*$.

Guess The attacker outputs a guess β' for β.

The advantage of an attacker is this game is defined to be $Pr[\beta = \beta'] - \frac{1}{2}$. We note that the model can easily be extended to handle chosen-ciphertext attacks by allowing for decryption queries in Phase 1 and Phase 2.

Definition 3 *A ciphertext-policy attribute-based encryption system is fully secure if all polynomial time attackers have at most a negligible advantage in this security game.*

Selective security is defined by adding an initialization phase where the attacker must declare $\mathbb{A}^*$ before seeing PK. Unlike previous works [5, 26, 41], we do not impose this restriction on the attacker.

2.1.4 Composite Order Bilinear Groups

We will construct our systems in composite order bilinear groups. Composite order bilinear groups were first introduced in [10]. We define a group generator $\mathcal{G}$, an algorithm which takes a security parameter λ as input and outputs a description of a bilinear group G. For our purposes, we will have $\mathcal{G}$ output $(p_1, p_2, p_3, G, G_T, e)$ where p_1, p_2, p_3 are distinct primes, G and G_T are cyclic groups of order $N = p_1 p_2 p_3$, and $e : G^2 \to G_T$ is a map such that:

1. (Bilinear) $\forall g, h \in G,\ a, b \in \mathbb{Z}_N,\ e(g^a, h^b) = e(g, h)^{ab}$
2. (Non-degenerate) $\exists g \in G$ such that $e(g, g)$ has order N in G_T.

We assume that the group operations in G and G_T as well as the bilinear map e are computable in polynomial time with respect to λ and that the group descriptions of G and G_T include generators of the respective cyclic groups. We let G_{p_1}, G_{p_2}, and G_{p_3} denote the subgroups of order p_1, p_2 and p_3 in G respectively. We note that when $h_i \in G_{p_i}$ and $h_j \in G_{p_j}$ for

$i \neq j$, $e(h_i, h_j)$ is the identity element in G_T. To see this, suppose $h_1 \in G_{p_1}$ and $h_2 \in G_{p_2}$. Let g denote a generator of G. Then, $g^{p_1 p_2}$ generates G_{p_3}, $g^{p_1 p_3}$ generates G_{p_2}, and $g^{p_2 p_3}$ generates G_{p_1}. Hence, for some α_1, α_2, $h_1 = (g^{p_2 p_3})^{\alpha_1}$ and $h_2 = (g^{p_1 p_3})^{\alpha_2}$. Then:

$$e(h_1, h_2) = e(g^{p_2 p_3 \alpha_1}, g^{p_1 p_3 \alpha_2}) = e(g^{\alpha_1}, g^{p_3 \alpha_2})^{p_1 p_2 p_3} = 1.$$

This orthogonality property of $G_{p_1}, G_{p_2}, G_{p_3}$ will be used to implement semi-functionality in our constructions.

We now state the complexity assumptions that we will rely on to prove security of our systems. These same assumptions were used by Lewko and Waters to obtain full security of their IBE and HIBE constructions in composite order groups [30]. With permission, we have reproduced the proof of the fact that these assumptions hold in the generic group model from [30] in Appendix A.2 for self-containment. We note that all three assumptions are static (constant size) and the first assumption is just the subgroup decision problem in the case where the group order is a product of three primes.

In the assumptions below, we let $G_{p_1 p_2}$, e.g., denote the subgroup of order $p_1 p_2$ in G.

Assumption 1 (Subgroup decision problem for 3 primes) Given a group generator $\mathcal{G}$, we define the following distribution:

$$\mathbb{G} = (N = p_1 p_2 p_3, G, G_T, e) \xleftarrow{R} \mathcal{G},$$

$$g \xleftarrow{R} G_{p_1},\ X_3 \xleftarrow{R} G_{p_3},$$

$$D = (\mathbb{G}, g, X_3),$$

$$T_1 \xleftarrow{R} G_{p_1 p_2},\ T_2 \xleftarrow{R} G_{p_1}.$$

We define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 1 to be:

$$Adv1_{\mathcal{G},\mathcal{A}}(\lambda) := \big| Pr[\mathcal{A}(D, T_1) = 1] - Pr[\mathcal{A}(D, T_2) = 1] \big|.$$

We note that T_1 can be written (uniquely) as the product of an element of G_{p_1} and an element of G_{p_2}. We refer to these elements as the "G_{p_1} part of T_1" and the "G_{p_2} part of T_1" respectively. We will use this terminology in our proofs.

Definition 4 *We say that $\mathcal{G}$ satisfies Assumption 1 if $Adv1_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ for any polynomial time algorithm $\mathcal{A}$.*

Assumption 2 Given a group generator $\mathcal{G}$, we define the following distribution:

$$\mathbb{G} = (N = p_1 p_2 p_3, G, G_T, e) \xleftarrow{R} \mathcal{G},$$

$$g, X_1 \xleftarrow{R} G_{p_1},\ X_2, Y_2 \xleftarrow{R} G_{p_2},\ X_3, Y_3 \xleftarrow{R} G_{p_3},$$

$$D = (\mathbb{G}, g, X_1 X_2, X_3, Y_2 Y_3),$$

$$T_1 \xleftarrow{R} G,\ T_2 \xleftarrow{R} G_{p_1 p_3}.$$

We define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 2 to be:

$$Adv2_{\mathcal{G},\mathcal{A}}(\lambda) := \big| Pr[\mathcal{A}(D, T_1) = 1] - Pr[\mathcal{A}(D, T_2) = 1] \big|.$$

We use $G_{p_1 p_3}$ to denote the subgroup of order $p_1 p_3$ in G. We note that T_1 can be (uniquely) written as the product of an element of G_{p_1}, an element of G_{p_2}, and an element of G_{p_3}. We refer to these as the "G_{p_1} part of T_1", the "G_{p_2} part of T_1", and the "G_{p_3} part of T_1", respectively. T_2 can similarly be written as the product of an element of G_{p_1} and an element of G_{p_3}.

Definition 5 *We say that $\mathcal{G}$ satisfies Assumption 2 if $Adv2_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ for any polynomial time algorithm $\mathcal{A}$.*

Assumption 3 Given a group generator $\mathcal{G}$, we define the following distribution:

$$\mathbb{G} = (N = p_1p_2p_3, G, G_T, e) \xleftarrow{R} \mathcal{G},\ \alpha, s \xleftarrow{R} \mathbb{Z}_N,$$

$$g \xleftarrow{R} G_{p_1},\ X_2, Y_2, Z_2 \xleftarrow{R} G_{p_2},\ X_3 \xleftarrow{R} G_{p_3},$$

$$D = (\mathbb{G}, g, g^{\alpha}X_2, X_3, g^{s}Y_2, Z_2),$$

$$T_1 = e(g,g)^{\alpha s},\ T_2 \xleftarrow{R} G_T.$$

We define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 3 to be:

$$Adv3_{\mathcal{G},\mathcal{A}}(\lambda) := \big|Pr[\mathcal{A}(D, T_1) = 1] - Pr[\mathcal{A}(D, T_2) = 1]\big|.$$

Definition 6 *We say that $\mathcal{G}$ satisfies Assumption 3 if $Adv3_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ for any polynomial time algorithm $\mathcal{A}$.*

2.2 Transformation from One-Use CP-ABE

Here we show how to obtain a fully secure CP-ABE system where attributes are used multiple times from a fully secure CP-ABE system where attributes are used only once. We do this with a simple encoding technique.

Suppose we have a CP-ABE system with a universe of n attributes with LSSS access structures that is secure when the function ρ is injective for each access structure associated to a ciphertext (i.e. attributes are only used once in the row labeling the of the share-generating matrix). Suppose we would like to have a system with n attributes where attributes can be used $\leq k$ times in the row labeling of a share-generating matrix. We can realize this by essentially taking k copies of each attribute in the system: instead of a single attribute B, we will have new "attributes" $B:1, \ldots, B:k$. Each time we want to label a row of an access matrix A with B, we label it with $B:i$ for a new value of i. We let ρ denote the original row labeling of A and ρ' denote this new row labeling. Each time we want to associate a subset S of attributes to a key, we instead use $S' := \{B:1, \ldots, B:k | B \in S\}$. We can then employ the one use system on the new universe of kn attributes and retain its full security. We note that the set S' satisfies the access structure (A, ρ') if and only if the set S satisfies the access structure (A, ρ).

For our construction, the sizes of the public parameters and the secret keys grow linearly in the number of involved attributes, so these will expand by a factor of k under this transformation. Note that the size of the access matrix does not change, so ciphertexts in our construction will remain the same size.

2.3 Our Fully Secure CP-ABE System

We construct our fully secure CP-ABE system in composite order groups of order $N = p_1p_2p_3$ with LSSS access structures. We note the strong resemblance between our system and the selectively secure CP-ABE system of Waters [41]. The KP-ABE system we give in Section A.1 also bears a strong resemblance to the selectively secure schemes in [26]. We thus provide additional examples of the phenomenon noted by [42, 30]: dual system encryption is a powerful and versatile tool for transforming selectively secure schemes into fully secure ones.

The normal operation of our system essentially occurs in the subgroup G_{p_1}. Keys are additionally randomized in G_{p_3}, and the subgroup G_{p_2} is our semi-functional space, which is

not used in the real system. Keys and ciphertexts will be semi-functional when they involve elements in the G_{p_2} subgroup. When normal keys are paired with semi-functional ciphertexts or semi-functional keys are paired with normal ciphertexts, the elements in G_{p_2} will not contribute to the pairings because they are orthogonal to elements in the G_{p_1} and G_{p_3} subgroups. When we pair a semi-functional key with a semi-functional ciphertext, we get an extra term arising from pairing the corresponding elements of G_{p_2} which will cause decryption to fail, unless this extra term happens to be zero. When this cancelation occurs and decryption still works, we say the key is *nominally* semi-functional. In other words, nominally semi-functional keys involve elements in G_{p_2}, but these cancel when paired with the G_{p_2} elements involved in the semi-functional ciphertext.

Our proof of security will rely on the restriction that each attribute can only be used once in the row labeling of an access matrix. This is because we will argue that a nominally semi-functional key is identically distributed to a regular semi-functional key in the attacker's view, since the attacker cannot ask for keys that can decrypt the challenge ciphertext. This information-theoretic argument fails when attributes can be used multiple times. Nonetheless, we can achieve full security for a system which uses attributes multiple times through the transformation given in the last section.

We believe that our fully secure system in composite order groups can be transformed to a fully secure system in prime order groups. This was accomplished for the previous applications of dual system encryption in [42, 30].

2.3.1 Construction

Setup$(\lambda, U) \rightarrow PK, MSK$ The setup algorithm chooses a bilinear group G of order $N = p_1 p_2 p_3$ (3 distinct primes). We let G_{p_i} denote the subgroup of order p_i in G. It then chooses random exponents $\alpha, a \in \mathbb{Z}_N$, and a random group element $g \in G_{p_1}$. For each attribute $i \in U$, it chooses a random value $s_i \in \mathbb{Z}_N$. The public parameters PK are $N, g, g^a, e(g,g)^\alpha, T_i = g^{s_i} \forall i$. The master secret key MSK is α and a generator X_3 of G_{p_3}.

KeyGen($MSK, S, PK) \rightarrow SK$ The key generation algorithm chooses a random $t \in \mathbb{Z}_N$, and random elements $R_0, R_0', R_i \in G_{p_3}$. The secret key is:

$$S,\ K = g^\alpha g^{at} R_0,\ L = g^t R_0',\ K_i = T_i^t R_i\ \forall i \in S.$$

Encrypt($(A, \rho), PK, M) \rightarrow CT$ A is an $\ell \times n$ matrix and ρ is map from each row A_x of A to an attribute $\rho(x)$. The encryption algorithm chooses a random vector $v \in \mathbb{Z}_N^n$, denoted $v = (s, v_2, \ldots, v_n)$. For each row A_x of A, it chooses a random $r_x \in \mathbb{Z}_N$. The ciphertext is (we also include (A, ρ) in the ciphertext, though we do not write it below):

$$C = M e(g,g)^{\alpha s},\ C' = g^s,$$

$$C_x = g^{a A_x \cdot v} T_{\rho(x)}^{-r_x},\ D_x = g^{r_x}\ \forall x.$$

Decrypt($CT, PK, SK) \rightarrow M$ The decryption algorithm computes constants $\omega_x \in \mathbb{Z}_N$ such that $\sum_{\rho(x) \in S} \omega_x A_x = (1, 0, \ldots, 0)$. It then computes:

$$e(C', K) / \prod_{\rho(x) \in S} \left(e(C_x, L) e(D_x, K_{\rho(x)}) \right)^{\omega_x} = e(g,g)^{\alpha s}.$$

Then M can be recovered as $C / e(g,g)^{\alpha s}$.

2.3.2 Security

Before we give our proof of security, we need to define two additional structures: semi-functional ciphertexts and keys. These will not be used in the real system, but will be needed in our proof.

Semi-functional Ciphertext A semi-functional ciphertext is formed as follows. We let g_2 denote a generator of G_{p_2} and c a random exponent modulo N. We also choose random values $z_i \in \mathbb{Z}_N$ associated to attributes, random values $\gamma_x \in \mathbb{Z}_N$ associated to matrix rows x, and a random vector $u \in \mathbb{Z}_N^n$. Then:

$$C' = g^s g_2^c,\ C_x = g^{aA_x \cdot v} T_{\rho(x)}^{-r_x} g_2^{A_x \cdot u + \gamma_x z_{\rho(x)}},\ D_x = g^{r_x} g_2^{-\gamma_x}\ \forall x.$$

Semi-functional Key A semi-functional key will take on one of two forms. A semi-functional key of type 1 is formed as follows. Exponents $t, d, b \in \mathbb{Z}_N$ and elements $R_0, R_0', R_i \in G_{p_3}$ are chosen randomly. The key is set as:

$$K = g^\alpha g^{at} R_0 g_2^d,\ L = g^t R_0' g_2^b,\ K_i = T_i^t R_i g_2^{bz_i}\ \forall i \in S.$$

A semi-functional key of type 2 is formed without the terms g_2^b and $g_2^{bz_i}$ (one could also interpret this as setting $b = 0$):

$$K = g^\alpha g^{at} R_0 g_2^d,\ L = g^t R_0',\ K_i = T_i^t R_i\ \forall i \in S.$$

We note that when we use a semi-functional key to decrypt a semi-functional ciphertext, we are left with an additional term:

$$e(g_2, g_2)^{cd - bu_1},$$

where u_1 denotes the first coordinate of u (i.e. $(1, 0, \ldots, 0) \cdot u$). We also note that these values z_i are common to semi-functional ciphertexts and semi-functional keys of type 1. These z_i terms always cancel when semi-functional keys are paired with semi-functional ciphertexts, so they do not hinder decryption. Instead, they are used as blinding factors to hide the value being shared in the G_{p_2} subgroup of a semi-functional ciphertext (the value u_1) from an attacker who cannot decrypt. This is where our one-use restriction is crucial: an attacker with a single semi-functional key of type 1 which cannot decrypt the challenge ciphertext should only be able to gain very limited information-theoretic knowledge of the z_i values. If attributes are used multiple times, too many z_i values may be exposed to the attacker. In each of the games we define below, at most one key is semi-functional of type 1 and all other semi-functional keys are type 2. This is to avoid information-theoretically leaking the z_i values by using them in multiple keys at once.

We call a semi-functional key of type 1 *nominally* semi-functional if $cd - bu_1 = 0$. Notice that when such a key is used to decrypt a corresponding semi-functional ciphertext, decryption will succeed.

We will prove the security of our system from Assumptions 1, 2, and 3 using a hybrid argument over a sequence of games. The first game, Game_{Real}, is the real security game (the ciphertext and all the keys are normal). In the next game, Game_0, all of the keys will be normal, but the challenge ciphertext will be semi-functional. We let q denote the number of key queries made by the attacker. For k from 1 to q, we define:

Game$_{k,1}$ In this game, the challenge ciphertext is semi-functional, the first $k-1$ keys are semi-functional of type 2, the k^{th} key is semi-functional of type 1, and the remaining keys are normal.

Game$_{k,2}$ In this game, the challenge ciphertext is semi-functional, the first k keys are semi-functional of type 2, and the remaining keys are normal.

We note that in Game$_{q,2}$, all of the keys are semi-functional of type 2. In the final game, Game$_{Final}$, all keys are semi-functional of type 2 and the ciphertext is a semi-functional encryption of a random message, independent of the two messages provided by the attacker. In Game$_{Final}$, the attacker's advantage is 0. We will prove these games are indistinguishable in the following four lemmas. For notational purposes in the lemmas below, we think of $Game_{0,2}$ as another way of denoting Game 0.

Lemma 7 *Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $Game_{Real}Adv_{\mathcal{A}} - Game_0Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 1.*

Proof. $\mathcal{B}$ is given g, X_3, T. It will simulate Game$_{Real}$ or Game$_0$ with $\mathcal{A}$. $\mathcal{B}$ chooses random exponents $a, \alpha \in \mathbb{Z}_N$ and a random exponent $s_i \in \mathbb{Z}_N$ for each attribute i in the system. It then sends $\mathcal{A}$ the public parameters:

$$PK = \{N,\ g,\ g^a,\ e(g,g)^\alpha,\ T_i = g^{s_i}\ \forall i\}.$$

It can generate normal keys in response to $\mathcal{A}$'s key requests by using the key generation algorithm, since it knows the $MSK = \{\alpha, X_3\}$.

$\mathcal{A}$ sends $\mathcal{B}$ two messages M_0, M_1 and an access matrix (A^*, ρ). To make the challenge ciphertext, $\mathcal{B}$ will implicitly set g^s to be the G_{p_1} part of T (we mean that T is the product of $g^s \in G_{p_1}$ and possibly an element of G_{p_2}). It chooses a random $\beta \in \{0,1\}$ and sets:

$$C = M_\beta e(g^\alpha, T),\ C' = T.$$

To form C_x for each row x of A^*, $\mathcal{B}$ first chooses random values $v'_2, \ldots, v'_n \in \mathbb{Z}_N$ and creates the vector $v' = (1, v'_2, \ldots, v'_n)$. It also chooses a random $r'_x \in \mathbb{Z}_N$. It sets:

$$C_x = T^{(aA_x \cdot v')} T^{-r'_x s_{\rho(x)}},\ D_x = T^{r'_x}.$$

We note that this implicitly sets $v = (s, sv'_2, \ldots, sv'_n)$ and $r_x = r'_x s$. Modulo p_1, this v is a random vector with first coordinate s and r_x is a random value. So if $T \in G_{p_1}$, this is a properly distributed normal ciphertext.

If $T \in G_{p_1 p_2}$, we let g_2^c denote the G_{p_2} part of T (i.e. $T = g^s g_2^c$). We then have a semi-functional ciphertext with $u = cav'$, $\gamma_x = -cr'_x$, and $z_{\rho(x)} = s_{\rho(x)}$. Though we are reusing values from the G_{p_1} parts here, this does not result in unwanted correlations. The values of $a, v'_2, \ldots, v'_n, r'_x, s_{\rho(x)}$ modulo p_2 are uncorrelated from their values modulo p_1 by the Chinese Remainder Theorem, so this is a properly distributed semi-functional ciphertext. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to break Assumption 1 with advantage ϵ. □

Lemma 8 *Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $Game_{k-1,2}Adv_{\mathcal{A}} - Game_{k,1}Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage negligibly close to ϵ in breaking Assumption 2.*

Proof. $\mathcal{B}$ is given $g, X_1X_2, X_3, Y_2Y_3, T$. It will simulate $\text{Game}_{k-1,2}$ or $\text{Game}_{k,1}$ with $\mathcal{A}$. It chooses random exponents $a, \alpha \in \mathbb{Z}_N$ and a random exponent $s_i \in \mathbb{Z}_N$ for each attribute i in the system. It then sends $\mathcal{A}$ the public parameters:

$$PK = \{N,\ g,\ g^a,\ e(g,g)^\alpha,\ T_i = g^{s_i}\ \forall i\}.$$

To make the first $k-1$ keys semi-functional of type 2, $\mathcal{B}$ responds to each key request by choosing a random $t \in \mathbb{Z}_N$, random elements R_0', R_i of G_{p_3}, and setting:

$$K = g^\alpha g^{at}(Y_2Y_3)^t,\ L = g^t R_0',\ K_i = T_i^t R_i\ \forall i \in S.$$

We note that K is properly distributed because the values of t modulo p_2 and p_3 are uncorrelated to its value modulo p_1. To make normal keys for requests $> k$, $\mathcal{B}$ can simply run the key generation algorithm since it knows the MSK.

To make key k, $\mathcal{B}$ will implicity set g^t equal to the G_{p_1} part of T. $\mathcal{B}$ chooses random elements R_0, R_0', R_i in G_{p_3} and sets:

$$K = g^\alpha T^a R_0,\ L = TR_0',\ K_i = T^{s_i} R_i\ \forall i \in S.$$

We note that if $T \in G_{p_1p_3}$, this is a properly distributed normal key. If $T \in G$, this is a semi-functional key of type 1. In this case, we have implicitly set $z_i = s_i$. If we let g_2^b denote the G_{p_2} part of T, we have that $d = ba$ modulo p_2 (i.e. the G_{p_2} part of K is $g_2^b a$, the G_{p_2} part of L is g_2^b, and the G_{p_2} part of K_i is $g_2^{bz_i}$. Note that the value of z_i modulo p_2 is uncorrelated from the value of s_i modulo p_1.

$\mathcal{A}$ sends $\mathcal{B}$ two messages M_0, M_1 and an access matrix (A^*, ρ). To make the semi-functional challenge ciphertext, $\mathcal{B}$ implicitly sets $g^s = X_1$ and $g_2^c = X_2$. It chooses random values $u_2, \ldots, u_n \in \mathbb{Z}_N$ and defines the vector u' as $u' = (a, u_2, \ldots, u_n)$. It also chooses a random exponent $r_x' \in \mathbb{Z}_N$. The ciphertext is formed as:

$$C = M_\beta e(g^\alpha, X_1X_2),\ C' = X_1X_2,\ C_x = (X_1X_2)^{A_x^* \cdot u'}(X_1X_2)^{-r_x' s_{\rho(x)}},\ D_x = (X_1X_2)^{r_x'}.$$

We note that this sets $v = sa^{-1}u'$ and $u = cu'$, so s is being shared in the G_{p_1} subgroup and ca is being shared in the G_{p_2} subgroup. This also implicitly sets $r_x = r_x's$, $\gamma_x = -cr_x'$. The values $z_{\rho(x)} = s_{\rho(x)}$ match those in the k^{th} key if it is semi-functional of type 1, as required.

The k^{th} key and ciphertext are *almost* properly distributed, except for the fact that the first coordinate of u (which equals ac) is correlated with the value of a modulo p_2 that also appears in key k if it is semi-functional. In fact, if the k^{th} key could decrypt the challenge ciphertext we would have $cd - bu_1 = cba - bca = 0$ modulo p_2, so our key is either normal or nominally semi-functional. We must argue that this is hidden to the attacker $\mathcal{A}$, who cannot request any keys that can decrypt the challenge ciphertext.

To argue that the value being shared in G_{p_2} in the challenge ciphertext is information-theoretically hidden, we appeal to our restriction that attributes are only used once in labeling the rows of the matrix. Since the k^{th} key cannot decrypt the challenge ciphertext, the rowspace R formed by the rows of the matrix whose attributes are in the key does not include the vector $(1, 0, \ldots, 0)$. (We may assume this holds modulo p_2 - if not, a non-trivial factor of N can be found, which breaks our complexity assumptions.) This means there is some vector w which is orthogonal to R and *not* orthogonal to $(1, 0, \ldots, 0)$ (modulo p_2). We fix a basis including w, and we can then write $u = fw + u''$ modulo p_2, where $f \in \mathbb{Z}_{p_2}$ and u'' is in the span of the basis elements not equal to w (and u'' is distributed uniformly randomly in this space). We note that u'' reveals no information about f, and that $u_1 = u \cdot (1, 0, \ldots, 0)$ cannot be determined from u'' alone - some information about f is needed, since w is not orthogonal to $(1, 0, \ldots, 0)$. However,

the shares corresponding to rows whose attributes are in the key only reveal information about u'', since w is orthogonal to R.

The only places fw appears are in equations of the form:

$$A_x^* \cdot u + \gamma_x z_{\rho(x)},$$

where the $\rho(x)$'s are each *unique* attributes not appearing the k^{th} key. As long as each γ_x is not congruent to 0 modulo p_2, each of these equations introduces a new unknown $z_{\rho(x)}$ that appears nowhere else, and so no information about f can be learned by the attacker. More precisely, for each potential value of u_1, there are an equal number of solutions to these equations, so each value is equally likely. Hence, the value being shared in the G_{p_2} subgroup in the semi-functional ciphertext is information-theoretically hidden, as long as each γ_x is non-zero modulo p_2. The probability that any of the γ_x values are congruent to 0 modulo p_2 is negligible. Thus, the ciphertext and key k are properly distributed in the attacker's view with probability negligibly close to 1.

Thus, if $T \in G_{p_1 p_3}$, then $\mathcal{B}$ has properly simulated $\text{Game}_{k-1,2}$, and if $T \in G$ and all the γ_x values are non-zero modulo p_2, then $\mathcal{B}$ has properly simulated $\text{Game}_{k,1}$. $\mathcal{B}$ can therefore use the output of $\mathcal{A}$ to gain advantage negligibly close to ϵ in breaking Assumption 2. $\square$

Lemma 9 *Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $Game_{k,1}Adv_{\mathcal{A}} - Game_{k,2}Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 2.*

Proof. This proof is very similar to the proof of the previous lemma, but the information-theoretic argument is no longer required. $\mathcal{B}$ is given $g, X_1X_2, X_3, Y_2Y_3, T$. It will simulate $\text{Game}_{k,1}$ or $\text{Game}_{k,2}$ with $\mathcal{A}$. It chooses random exponents $a, \alpha \in \mathbb{Z}_N$ and a random exponent $s_i \in \mathbb{Z}_N$ for each attribute i in the system. It then sends $\mathcal{A}$ the public parameters:

$$PK = \{N,\ g,\ g^a,\ e(g,g)^\alpha,\ T_i = g^{s_i}\ \forall i\}.$$

The $k-1$ semi-functional keys of type 2, the normal keys $> k$, and the challenge ciphertext are constructed exactly as in the previous lemma. This means the ciphertext is sharing the value ac in the G_{p_2} subgroup. This time, this will not be correlated with key k in any way, so this value is random modulo p_2 (note that a modulo p_1 and a modulo p_2 are uncorrelated).

To make key k, we proceed as we did before, but we additionally choose a random exponent $h \in \mathbb{Z}_N$ and set:

$$K = g^\alpha T^a R_0 (Y_2Y_3)^h,\ L = TR_0',\ K_i = T^{s_i} R_i\ \forall i \in S.$$

The only change we have made here is adding the $(Y_2Y_3)^h$ term. This randomizes the G_{p_2} part of K, so the key is no longer nominally semi-functional. If we tried to decrypt the semi-functional ciphertext with it, decryption would fail (we no longer have the cancelation $cd - bu_1 \equiv 0 (\text{mod}\, p_2)$).

If $T \in G_{p_1 p_3}$, this is a properly distributed semi-functional key of type 2. If $T \in G$, this is a properly distributed semi-functional key of type 1. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to gain advantage ϵ in breaking Assumption 2. $\square$

Lemma 10 *Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $Game_{q,2}Adv_{\mathcal{A}} - Game_{Final}Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 3.*

	size of PK	size of SK	size of CT
one-use	$\|U\| + 3$	$\|S\| + 2$	$2\ell + 2$
k-use	$k\|U\| + 3$	$k\|S\| + 3$	$2\ell + 2$

Table 1: Comparison of one-use and k-use CP-ABE systems

Proof. Again, this proof is similar to the proofs of the previous lemmas. $\mathcal{B}$ is given g, $g^\alpha X_2$, X_3, $g^s Y_2$, Z_2, T. It will simulate $\text{Game}_{q,2}$ or Game_{Final} with $\mathcal{A}$. It chooses a random exponent $a \in \mathbb{Z}_N$ and a random exponent $s_i \in \mathbb{Z}_N$ for each attribute i in the system. It takes α from the assumption term $g^\alpha X_2$. It then sends $\mathcal{A}$ the public parameters:

$$PK = \{N,\ g,\ g^a,\ e(g,g)^\alpha = e(g, g^\alpha X_2),\ T_i = g^{s_i}\ \forall i\}.$$

To make the semi-functional keys of type 2, $\mathcal{B}$ responds to each key request by choosing a random $t \in \mathbb{Z}_N$, random elements R_0, R_0', R_i of G_{p_3}, and setting:

$$K = g^\alpha g^{at} Z_2^t R_0,\ L = g^t R_0',\ K_i = T_i^t R_i\ \forall i \in S,$$

as in the previous lemmas.

$\mathcal{A}$ sends $\mathcal{B}$ two messages M_0, M_1 and an access matrix (A^*, ρ). To make the semi-functional challenge ciphertext, $\mathcal{B}$ will take s from the assumption term $g^s Y_2$. It chooses random values $u_2, \ldots, u_n \in \mathbb{Z}_N$ and defines the vector u' as $u' = (a, u_2, \ldots, u_n)$. It also chooses a random exponent $r_x' \in \mathbb{Z}_N$. The ciphertext is formed as:

$$C = M_\beta T,\ C' = g^s Y_2,\ C_x = (g^s Y_2)^{A_x^* \cdot u'} (g^s Y_2)^{-r_x' s_{\rho(x)}},\ D_x = (g^s Y_2)^{r_x'}.$$

We note that this sets $v = sa^{-1}u'$ and $u = cu'$, so s is being shared in the G_{p_1} subgroup and ca is being shared in the G_{p_2} subgroup. This also implicitly sets $r_x = r_x' s$, $\gamma_x = -cr_x'$.

If $T = e(g,g)^{\alpha s}$, this is a properly distributed semi-functional encryption of M_b. Otherwise, it is a properly distributed semi-functional encryption of a random message in G_T. Thus, $\mathcal{B}$ can use the output $\mathcal{A}$ to gain advantage ϵ in breaking Assumption 3. □

We have now proven the following theorem:

Theorem 11 *If Assumptions 1, 2, and 3 hold, then our CP-ABE system is secure.*

Proof. If Assumptions 1, 2, and 3 hold, then we have shown by the previous lemmas that the real security game is indistinguishable from Game_{Final}, in which the value of β is information-theoretically hidden from the attacker. Hence the attacker cannot attain a non-negligible advantage in breaking the CP-ABE system. □

2.3.3 Expanding to Multi-Use

To build a fully secure CP-ABE system where each attribute can be used up to k times in the row labeling of an access matrix, we apply the encoding technique of Section 2.2. We compare the resulting k-use system to the one-use system in Table 1. Note that the encoding does not increase the size of the ciphertext. We let $|U|$ denote the number of attributes before the encoding is applied, $|S|$ denote the number of attributes associated with a key, and ℓ denote the number of rows in an access matrix associated with a ciphertext. All sizes refer to the number of group elements.

2.4 Discussion

We have obtained the first fully secure CP-ABE system in the standard model. Our techniques also yield a fully secure KP-ABE system. Our KP-ABE system and the proof of its security can be found in Section A.1. Essentially, a KP-ABE system is like a CP-ABE system with the roles of keys and ciphertexts reversed: in a KP-ABE system, keys are associated with access structures and ciphertexts are associated with subsets of attributes. Our techniques readily adapt to KP-ABE, and the proof of security is very similar to the CP-ABE case.

It is also possible to adapt our techniques to obtain a large universe construction. In our current construction, the size of the public parameters is linear in the number of attributes in the universe. In a large universe construction, we could use all elements of $\mathbb{Z}_{p_1}^*$ as attributes, with the size of the public parameters linear in n, a parameter which denotes the maximum size of a set of attributes used in the system. This reduces the size of the public parameters and allows us to use arbitrary strings as attributes by applying a collision-resistant hash function $H : \{0,1\}^* \to \mathbb{Z}_{p_1}^*$. Note that these attributes no longer need to have been considered during setup. To obtain a large universe construction, we could replace the group elements T_i associated with attributes i with a function $T : \mathbb{Z}_{p_1} \to G_{p_1}$ based on a degree n polynomial. Goyal, Pandey, Sahai, and Waters [26] do this for their KP-ABE construction.

Though we build our ABE systems in composite order bilinear groups, we believe that similar systems can be constructed in prime order groups. Composite order groups are a natural setting for applications of dual system encryption, since orthogonality of subgroups under pairings allows us to associate our semi-functional space with a particular subgroup. This is very convenient and leads to relatively simple fully secure schemes which closely resemble previously known selectively secure schemes. Though composite order groups have these advantages, they are not necessary for the dual system encryption technique.

Waters [42] first instantiated his fully secure IBE and HIBE systems in composite order groups and then transferred them into prime order groups, obtaining full security under the well-established $d - BDH$ and decisional Linear assumptions. Lewko and Waters [30] built upon these ideas to obtain an analog of their IBE system in asymmetric prime order groups. The introduction of asymmetry simplified their construction, at the expense of relying on non-standard (static) assumptions. Freeman [21] also discusses a general class of transformations from composite order groups to prime order groups, but this does not encompass our construction. In the future, these transformation techniques might be extended to obtain versions of our ABE schemes in prime order groups.

3 Fully Secure Predicate Encryption

3.1 Our Approach and Key Technique

3.1.1 Dual Pairing Vector Spaces (DPVS)

We now briefly explain our approach, DPVS, constructed on symmetric pairing groups $(q, \mathbb{G}, \mathbb{G}_T, g, e)$, where q is a prime, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order q, g is a generator of $\mathbb{G}$, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a non-degenerate bilinear pairing operation, and $g_T := e(g, g) \neq 1$. Here we denote the group operation of $\mathbb{G}$ and $\mathbb{G}_T$ by multiplication. Note that this construction also works on *asymmetric* pairing groups (in this paper, we use symmetric pairing groups for simplicity of description). As for the definitions of some notations, see the last part of this subsection.

Vector space $\mathbb{V}$: $\mathbb{V} := \overbrace{\mathbb{G} \times \cdots \times \mathbb{G}}^{N}$, whose element is expressed by N-dimensional vector, $\boldsymbol{x} := (g^{x_1}, \ldots, g^{x_N})$ ($x_i \in \mathbb{F}_q$ for $i = 1, \ldots, N$).

Canonical base $\mathbb{A}$: $\mathbb{A} := (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N)$ of $\mathbb{V}$, where $\boldsymbol{a}_1 := (g, 1, \ldots, 1)$, $\boldsymbol{a}_2 := (1, g, 1, \ldots, 1), \ldots,$ $\boldsymbol{a}_N := (1, \ldots, 1, g)$.

Pairing operation: $e(\boldsymbol{x}, \boldsymbol{y}) := \prod_{i=1}^{N} e(g^{x_i}, g^{y_i}) = e(g,g)^{\sum_{i=1}^{N} x_i y_i} = g_T^{\overrightarrow{x} \cdot \overrightarrow{y}} \in \mathbb{G}_T$, where $\boldsymbol{x} := (g^{x_1}, \ldots, g^{x_N}) = x_1\boldsymbol{a}_1 + \cdots + x_N\boldsymbol{a}_N \in \mathbb{V}$, $\boldsymbol{y} := (g^{y_1}, \ldots, g^{y_N}) = y_1\boldsymbol{a}_1 + \cdots + y_N\boldsymbol{a}_N \in \mathbb{V}$, $\overrightarrow{x} := (x_1, \ldots, x_N)$ and $\overrightarrow{y} := (y_1, \ldots, y_N)$. Here, $\boldsymbol{x}$ and $\boldsymbol{y}$ can be expressed by coefficient vector over basis $\mathbb{A}$ such that $(x_1, \ldots, x_N)_{\mathbb{A}} = (\overrightarrow{x})_{\mathbb{A}} := \boldsymbol{x}$ and $(y_1, \ldots, y_N)_{\mathbb{A}} = (\overrightarrow{y})_{\mathbb{A}} := \boldsymbol{y}$.

Base change: Canonical basis $\mathbb{A}$ is changed to basis $\mathbb{B} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_N)$ of $\mathbb{V}$ using a uniformly chosen (regular) linear transformation, $X := (\chi_{i,j}) \xleftarrow{\cup} GL(N, \mathbb{F}_q)$, such that $\boldsymbol{b}_i = \sum_{j=1}^{N} \chi_{i,j}\boldsymbol{a}_j$, $(i = 1, \ldots, N)$. $\mathbb{A}$ is also changed to basis $\mathbb{B}^* := (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_N^*)$ of $\mathbb{V}$, such that $(\vartheta_{i,j}) := (X^T)^{-1}$, $\boldsymbol{b}_i^* = \sum_{j=1}^{N} \vartheta_{i,j}\boldsymbol{a}_j$, $(i = 1, \ldots, N)$. We see that $e(\boldsymbol{b}_i, \boldsymbol{b}_j^*) = g_T^{\delta_{i,j}}$, ($\delta_{i,j} = 1$ if $i = j$, and $\delta_{i,j} = 0$ if $i \neq j$) i.e., $\mathbb{B}$ and $\mathbb{B}^*$ are dual orthonormal bases of $\mathbb{V}$.

Here, $\boldsymbol{x} := x_1\boldsymbol{b}_1 + \cdots + x_N\boldsymbol{b}_N \in \mathbb{V}$ and $\boldsymbol{y} := y_1\boldsymbol{b}_1^* + \cdots + y_N\boldsymbol{b}_N^* \in \mathbb{V}$ can be expressed by coefficient vectors over $\mathbb{B}$ and $\mathbb{B}^*$ such that $(x_1, \ldots, x_N)_{\mathbb{B}} = (\overrightarrow{x})_{\mathbb{B}} := \boldsymbol{x}$ and $(y_1, \ldots, y_N)_{\mathbb{B}^*} = (\overrightarrow{y})_{\mathbb{B}^*} := \boldsymbol{y}$, and $e(\boldsymbol{x}, \boldsymbol{y}) = e(g,g)^{\sum_{i=1}^{N} x_i y_i} = g_T^{\overrightarrow{x} \cdot \overrightarrow{y}} \in \mathbb{G}_T$.

Intractable problem: One of the most natural decisional problems in this approach is the decisional subspace problem [32]. It is to distinguish $\boldsymbol{v} := v_{N_2+1}\boldsymbol{b}_{N_2+1} + \cdots + v_{N_1}\boldsymbol{b}_{N_1}$ $(= (0, \ldots, 0, v_{N_2+1}, \ldots, v_{N_1})_{\mathbb{B}})$, from $\boldsymbol{u} := v_1\boldsymbol{b}_1 + \cdots + v_{N_1}\boldsymbol{b}_{N_1}$ $(= (v_1, \ldots, v_{N_1})_{\mathbb{B}})$, where $(v_1, \ldots, v_{N_1}) \xleftarrow{\cup} \mathbb{F}_q^{N_1}$ and $N_2 + 1 < N_1$.

Trapdoor: Although the decisional subspace problem is assumed to be intractable, it can be efficiently solved by using *trapdoor* $\boldsymbol{t}^* \in \mathsf{span}\langle \boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_{N_2}^* \rangle$. Given $\boldsymbol{v} := v_{N_2+1}\boldsymbol{b}_{N_2+1} + \cdots + v_{N_1}\boldsymbol{b}_{N_1}$ or $\boldsymbol{u} := v_1\boldsymbol{b}_1 + \cdots + v_{N_1}\boldsymbol{b}_{N_1}$, we can distinguish $\boldsymbol{v}$ from $\boldsymbol{u}$ using $\boldsymbol{t}^*$ since $e(\boldsymbol{v}, \boldsymbol{t}^*) = 1$ and $e(\boldsymbol{u}, \boldsymbol{t}^*) \neq 1$ with high probability.

Advantage of this approach: Higher dimensional vector treatment of bilinear pairing groups have been already employed in literature especially in the areas of IBE, ABE and BE (e.g., [8, 5, 11, 14, 26, 27, 36]). For example, in a typical vector treatment, two vector forms of $P := (g^{x_1}, \ldots, g^{x_N})$ and $Q := (g^{y_1}, \ldots, g^{y_N})$ are set and pairing for P and Q is operated as $e(P, Q) := \prod_{i=1}^{N} e(g^{x_i}, g^{y_i})$. Such treatment can be rephrased in this approach such that $P = x_1\boldsymbol{a}_1 + \cdots + x_N\boldsymbol{a}_N$ $(= (x_1, \ldots, x_N)_{\mathbb{A}})$, and $Q = y_1\boldsymbol{a}_1 + \cdots + y_N\boldsymbol{a}_N$ $(= (y_1, \ldots, y_N)_{\mathbb{A}})$ over canonical basis $\mathbb{A}$.

The major drawback of this approach is the easily *decomposable* property over $\mathbb{A}$ (i.e., the decisional subspace problem is easily solved). That is, it is easy to decompose $x_i\boldsymbol{a}_i = (1, \ldots, 1, g^{x_i}, 1, \ldots, 1)$ from $P := x_1\boldsymbol{a}_1 + \cdots x_N\boldsymbol{a}_N = (g^{x_1}, \ldots, g^{x_N})$.

In contrast, our approach employs basis $\mathbb{B}$, which is linearly transformed from $\mathbb{A}$ using a secret random matrix $X \in \mathbb{F}_q^{N \times N}$. A remarkable property over $\mathbb{B}$ is that it seems hard to decompose $x_i\boldsymbol{b}_i$ from $P' := x_1\boldsymbol{b}_1 + \cdots x_N\boldsymbol{b}_N$ (and the decisional subspace problem seems intractable). In addition, the secret matrix X (and the dual orthonormal basis $\mathbb{B}^*$ of $\mathbb{V}$) can be used as a source of the trapdoors to the decomposability (and distinguishability for the decisional subspace problem through the pairing operation over $\mathbb{B}$ and $\mathbb{B}^*$ as mentioned above). The hard decomposability (and indistinguishability) and its trapdoors are ones of the key tricks in this paper. In addition, a part of basis $\mathbb{B}$ can be hidden to users (and adversaries), and the hidden part of $\mathbb{B}$ plays a key role in our security proofs (e.g., $(\boldsymbol{b}_{n+1}, \ldots, \boldsymbol{b}_{2n})$ are hidden to users in the proposed PE scheme). Note that composite order pairing groups are often employed with similar tricks such as hard decomposability

(and indistinguishability) of a composite order group to the prime order subgroups and its trapdoors through factoring (e.g., [29, 38]).

3.1.2 Dual System Encryption Methodology

At the top level of strategy of the security proof, we follow the dual system encryption methodology proposed by Waters [42]. Security is proven using a sequence of games. Game 0 is the real security game. In Game 1, the target ciphertext is changed to semi-functional. When ν secret key queries are issued by an adversary, there are ν game changes from Game 1 (Game 2-0) through Game 2-ν. In Game 2-k, the first k keys are semi-functional while the remaining keys are normal. The final game with advantage 0 is changed from Game 2-ν. As usual, we prove that the advantage gaps between neighboring games are negligible.

The most difficult part in the security proof, *especially for inner-product predicate encryption*, is how to resolve a paradoxical problem to prove the negligible gap between Game 2-k and Game 2-$(k-1)$, where the simulator (for the security proof) itself may distinguish the simulated k-th key (semi-functional key) in Game 2-k and the k-th key (normal key) in Game 2-$(k-1)$ by using a simulated (semi-functional) ciphertext, since the simulator can make ciphertexts and keys for any legal attributes and predicates (especially, in the adaptive security game, the simulator should generate a target ciphertext associated with any attribute adaptively selected by the adversary).

For (H)IBE, this problem was resolved by introducing tricks such that the simulated k-th key and ciphertext have a special correlation regarding the equality of their identity values [30, 42].

This problem is much harder for inner-product predicate encryption. Given a predicate vector $\overrightarrow{v}$ for secret key $\mathsf{sk}_{\overrightarrow{v}}$, there are exponentially many (orthogonal) attribute vectors $\overrightarrow{x}$ for ciphertext $c_{\overrightarrow{x}}$ such that $\mathsf{sk}_{\overrightarrow{v}}$ can decrypt $c_{\overrightarrow{x}}$, i.e., $\overrightarrow{v} \cdot \overrightarrow{x} = 0$. Therefore, in order to resolve the above-mentioned paradoxical problem, we should give some trick on the simulated k-th key $\mathsf{sk}_{\overrightarrow{v}}$ with $\overrightarrow{v}$ and all ciphertexts with $\overrightarrow{x}$ satisfying $\overrightarrow{v} \cdot \overrightarrow{x} = 0$, while a trick on the simulated k-th key sk_I with identity I and ciphertext with the same I is enough for (H)IBE.

We use *special form of semi-functional* keys and ciphertexts for simulating the k-th key and target ciphertext such that the simulated k-th key (a special form of semi-functional key) $\mathsf{sk}_{\overrightarrow{v}}$ in Game 2-k can decrypt *all* simulated ciphertexts (a special form of semi-functional ciphertexts) $c_{\overrightarrow{x}}$ with $\overrightarrow{x}$ satisfying $\overrightarrow{v} \cdot \overrightarrow{x} = 0$. Essentially, we adapt the notion of *nominally semi-functional* keys and ciphertexts that was introduced by Lewko and Waters [30] to the setting of inner product encryption.

In addition, the distribution of a pair comprising the simulated k-th key $\mathsf{sk}_{\overrightarrow{v}}$ and simulated ciphertext $c_{\overrightarrow{x}}$ (i.e., a *special semi-functional* key and ciphertext) is equivalent to that of an independent and random *semi-functional* key and ciphertext except with negligible probability, when $\overrightarrow{v} \cdot \overrightarrow{x} \neq 0$.

That is, the special forms of semi-functional keys and ciphertexts are correlated (for the case of $\overrightarrow{v} \cdot \overrightarrow{x} = 0$), but the adversary cannot notice the correlation since the adversary's queries should satisfy the condition $\overrightarrow{v} \cdot \overrightarrow{x} \neq 0$. In other words, nominal semi-functionality is information-theoretically hidden from the adversary. A more detailed explanation of how this is implemented on DPVS will be given in the proof outline in Section 3.5.2.

3.1.3 Notations

When A is a random variable or distribution, $y \xleftarrow{\mathsf{R}} A$ denotes that y is randomly selected from A according to its distribution. When A is a set, $y \xleftarrow{\mathsf{U}} A$ denotes that y is uniformly selected from A. $y := z$ denotes that y is set, defined or substituted by z. When a is a fixed value,

$A(x) \to a$ (e.g., $A(x) \to 1$) denotes the event that machine (algorithm) A outputs a on input x. A function $f : \mathbb{N} \to \mathbb{R}$ is *negligible* in λ, if for every constant $c > 0$, there exists an integer n such that $f(\lambda) < \lambda^{-c}$ for all $\lambda > n$.

We denote the finite field of order q by $\mathbb{F}_q$. A vector symbol denotes a vector representation over $\mathbb{F}_q$, e.g., $\overrightarrow{x}$ denotes $(x_1, \ldots, x_n) \in \mathbb{F}_q^n$. For two vectors $\overrightarrow{x} = (x_1, \ldots, x_n)$ and $\overrightarrow{v} = (v_1, \ldots, v_n)$, $\overrightarrow{x} \cdot \overrightarrow{v}$ denotes the inner-product $\sum_{i=1}^n x_i v_i$. X^{T} denotes the transpose of matrix X. I_ℓ and 0_ℓ denote the $\ell \times \ell$ identity matrix and the $\ell \times \ell$ zero matrix, respectively. A bold face letter denotes an element of vector space $\mathbb{V}$, e.g., $\boldsymbol{x} \in \mathbb{V}$. When $\boldsymbol{b}_i \in \mathbb{V}$ $(i = 1, \ldots, n)$, $\mathsf{span}\langle \boldsymbol{b}_1, \ldots, \boldsymbol{b}_n \rangle \subseteq \mathbb{V}$ (resp. $\mathsf{span}\langle \overrightarrow{x}_1, \ldots, \overrightarrow{x}_n \rangle$) denotes the subspace generated by $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ (resp. $\overrightarrow{x}_1, \ldots, \overrightarrow{x}_n$). For bases $\mathbb{B} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_N)$ and $\mathbb{B}^* := (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_N^*)$, $(x_1, \ldots, x_N)_{\mathbb{B}} := \sum_{i=1}^N x_i \boldsymbol{b}_i$ and $(y_1, \ldots, y_N)_{\mathbb{B}^*} := \sum_{i=1}^N y_i \boldsymbol{b}_i^*$.

3.2 Dual Pairing Vector Spaces by Direct Product of Symmetric Pairing Groups

Definition 12 *"Symmetric bilinear pairing groups"* $(q, \mathbb{G}, \mathbb{G}_T, g, e)$ *are a tuple of a prime* q*, cyclic (multiplicative) groups* $\mathbb{G}$ *and* $\mathbb{G}_T$ *of order* q*,* $g \neq 1 \in \mathbb{G}$*, and a polynomial-time computable nondegenerate bilinear pairing* $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ *i.e.,* $e(g^s, g^t) = e(g, g)^{st}$ *and* $e(g, g) \neq 1$*.*

Let $\mathcal{G}_{\mathsf{bpg}}$ *be an algorithm that takes input* 1^λ *and outputs a description of bilinear pairing groups* $(q, \mathbb{G}, \mathbb{G}_T, g, e)$ *with security parameter* λ*.*

In this paper, we concentrate on the symmetric version of dual pairing vector spaces [32, 33] constructed by using symmetric bilinear pairing groups given in Definition 12.

Definition 13 *"Dual pairing vector spaces (DPVS)"* $(q, \mathbb{V}, \mathbb{G}_T, \mathbb{A}, e)$ *by a direct product of symmetric pairing groups* $(q, \mathbb{G}, \mathbb{G}_T, g, e)$ *are a tuple of prime* q*,* N*-dimensional vector space* $\mathbb{V} := \overbrace{\mathbb{G} \times \cdots \times \mathbb{G}}^{N}$ *over* $\mathbb{F}_q$*, cyclic group* $\mathbb{G}_T$ *of order* q*, canonical basis* $\mathbb{A} := (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N)$ *of* $\mathbb{V}$*, where* $\boldsymbol{a}_i := (\overbrace{1, \ldots, 1}^{i-1}, g, \overbrace{1, \ldots, 1}^{N-i})$*, and pairing* $e : \mathbb{V} \times \mathbb{V} \to \mathbb{G}_T$*.*

The pairing is defined by $e(\boldsymbol{x}, \boldsymbol{y}) := \prod_{i=1}^N e(g_i, h_i) \in \mathbb{G}_T$ *where* $\boldsymbol{x} := (g_1, \ldots, g_N) \in \mathbb{V}$ *and* $\boldsymbol{y} := (h_1, \ldots, h_N) \in \mathbb{V}$*. This is nondegenerate bilinear i.e.,* $e(s\boldsymbol{x}, t\boldsymbol{y}) = e(\boldsymbol{x}, \boldsymbol{y})^{st}$ *and if* $e(\boldsymbol{x}, \boldsymbol{y}) = 1$ *for all* $\boldsymbol{y} \in \mathbb{V}$*, then* $\boldsymbol{x} = \boldsymbol{0}$*. For all* i *and* j*,* $e(\boldsymbol{a}_i, \boldsymbol{a}_j) = g_T^{\delta_{i,j}}$ *where* $\delta_{i,j} = 1$ *if* $i = j$*, and* 0 *otherwise, and* $g_T := e(g, g) \neq 1 \in \mathbb{G}_T$*.*

DPVS also has linear transformations $\phi_{i,j}$ *on* $\mathbb{V}$ *s.t.* $\phi_{i,j}(\boldsymbol{a}_j) = \boldsymbol{a}_i$ *and* $\phi_{i,j}(\boldsymbol{a}_k) = \boldsymbol{0}$ *if* $k \neq j$*, which can be easily achieved by* $\phi_{i,j}(\boldsymbol{x}) := (\overbrace{1, \ldots, 1}^{i-1}, g_j, \overbrace{1, \ldots, 1}^{N-i})$ *where* $\boldsymbol{x} := (g_1, \ldots, g_N)$*. We call* $\phi_{i,j}$ *"distortion maps".*

DPVS generation algorithm $\mathcal{G}_{\mathsf{dpvs}}$ *takes input* 1^λ $(\lambda \in \mathbb{N})$ *and* $N \in \mathbb{N}$*, and outputs a description of* $\mathsf{param}_{\mathbb{V}} := (q, \mathbb{V}, \mathbb{G}_T, \mathbb{A}, e)$ *with security parameter* λ *and* N*-dimensional* $\mathbb{V}$*. It can be constructed by using* $\mathcal{G}_{\mathsf{bpg}}$*.*

For the asymmetric version of DPVS, $(q, \mathbb{V}, \mathbb{V}^*, \mathbb{G}_T, \mathbb{A}, \mathbb{A}^*, e)$, see Appendix B.1. The above symmetric version is obtained by identifying $\mathbb{V} = \mathbb{V}^*$ and $\mathbb{A} = \mathbb{A}^*$ in the asymmetric version. (For the other realization using higher genus Jacobians, see [32].)

We describe random dual orthonormal bases generator $\mathcal{G}_{\mathsf{ob}}$ below, which is used as a sub-

routine in the proposed (H)PE scheme.

$$
\begin{aligned}
&\mathcal{G}_{\mathsf{ob}}(1^\lambda, N):\ \mathsf{param}_{\mathbb{V}} := (q, \mathbb{V}, \mathbb{G}_T, \mathbb{A}, e) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{dpvs}}(1^\lambda, N),\\
&\quad X := (\chi_{i,j}) \xleftarrow{\mathsf{U}} GL(N, \mathbb{F}_q),\ (\vartheta_{i,j}) := (X^{\mathrm{T}})^{-1},\\
&\quad \boldsymbol{b}_i := \textstyle\sum_{j=1}^{N} \chi_{i,j} \boldsymbol{a}_j,\ \mathbb{B} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_N),\ \boldsymbol{b}^*_i := \textstyle\sum_{j=1}^{N} \vartheta_{i,j} \boldsymbol{a}_j,\ \mathbb{B}^* := (\boldsymbol{b}^*_1, \ldots, \boldsymbol{b}^*_N),\\
&\quad \text{return}\ \ (\mathsf{param}_{\mathbb{V}}, \mathbb{B}, \mathbb{B}^*).
\end{aligned}
$$

3.3 Assumption

Definition 14 (n-eDDH: n-Extended Decisional Diffie-Hellman Assumption) *The n-eDDH problem is to guess $\beta \in \{0,1\}$, given $(\mathsf{param}_{\mathbb{G}},\ g, g^\kappa, \{g^{\omega+\gamma_i h_i}, g^{\gamma_i}, g^{h_i}\}_{1\le i\le n},\ \{g^{\gamma_i h_j}\}_{1\le i\neq j\le n},\ Y_\beta) \xleftarrow{\mathsf{R}} \mathcal{G}^{n\text{-}\mathsf{eDDH}}_\beta(1^\lambda)$, where*

$$
\begin{aligned}
&\mathcal{G}^{n\text{-}\mathsf{eDDH}}_\beta(1^\lambda):\ \mathsf{param}_{\mathbb{G}} := (q, \mathbb{G}, \mathbb{G}_T, g, e) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{bpg}}(1^\lambda),\\
&\quad \kappa \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times,\quad \omega, h_i, \gamma_i \xleftarrow{\mathsf{U}} \mathbb{F}_q \ \text{ for } i = 1, \ldots, n,\\
&\quad Y_0 := g^{\kappa\omega},\quad Y_1 \xleftarrow{\mathsf{U}} \mathbb{G},\\
&\quad \text{return}\ \ (\mathsf{param}_{\mathbb{G}},\ g, g^\kappa, \{g^{\omega+\gamma_i h_i}, g^{\gamma_i}, g^{h_i}\}_{1\le i\le n},\quad \{g^{\gamma_i h_j}\}_{1\le i\neq j\le n},\ Y_\beta),
\end{aligned}
$$

for $\beta \xleftarrow{\mathsf{U}} \{0,1\}$. For a probabilistic machine $\mathcal{C}$, we define the advantage of $\mathcal{C}$ for the n-eDDH problem as:

$$
\begin{aligned}
\mathsf{Adv}^{n\text{-}\mathsf{eDDH}}_{\mathcal{C}}(\lambda) := \ &\Big|\mathsf{Pr}\left[\mathcal{C}(1^\lambda, \varrho) \to 1 \,\middle|\, \varrho \xleftarrow{\mathsf{R}} \mathcal{G}^{n\text{-}\mathsf{eDDH}}_0(1^\lambda)\right]\\
&-\mathsf{Pr}\left[\mathcal{C}(1^\lambda, \varrho) \to 1 \,\middle|\, \varrho \xleftarrow{\mathsf{R}} \mathcal{G}^{n\text{-}\mathsf{eDDH}}_1(1^\lambda)\right]\Big|.
\end{aligned}
$$

The n-eDDH assumption is: For any polynomial-time adversary $\mathcal{C}$, the advantage $\mathsf{Adv}^{n\text{-}\mathsf{eDDH}}_{\mathcal{C}}(\lambda)$ is negligible.

The following lemma shows that the n-eDDH assumption is true in the generic bilinear pairing group model [8].

Lemma 15 *For any adversary $\mathcal{C}$ that makes a total of at most ν queries to the oracles computing the group operation in $\mathbb{G}$ and the bilinear pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, the advantage $\mathsf{Adv}^{n\text{-}\mathsf{eDDH}}_{\mathcal{C}}(\lambda)$ is $O((\nu + n^2)^2/2^\lambda)$ in the generic bilinear pairing group model.*

The proof of Lemma 15 is given in Appendix B.2.

3.4 Definition of Predicate Encryption

This section defines predicate encryption (PE) for the class of inner-product predicates and its security.

An attribute of inner-product predicates is expressed as a vector $\overrightarrow{x} \in \mathbb{F}_q^n \setminus \{\overrightarrow{0}\}$ and a predicate $f_{\overrightarrow{v}}$ is associated with a vector $\overrightarrow{v}$, where $f_{\overrightarrow{v}}(\overrightarrow{x}) = 1$ iff $\overrightarrow{v} \cdot \overrightarrow{x} = 0$. Let $\Sigma := \mathbb{F}_q^n \setminus \{\overrightarrow{0}\}$, i.e., the set of the attributes, and $\mathcal{F} := \{f_{\overrightarrow{v}} | \overrightarrow{v} \in \mathbb{F}_q^n \setminus \{\overrightarrow{0}\}\}$ i.e., the set of the predicates.

Definition 16 *A predicate encryption (PE) scheme for the class of inner-product predicates $\mathcal{F}$ and attributes Σ consists of probabilistic polynomial-time algorithms* Setup, KeyGen, Enc *and* Dec. *They are given as follows:*

- Setup *takes as input security parameter* 1^λ *outputs (master) public key* pk *and (master) secret key* sk.
- KeyGen *takes as input the master public key* pk*, secret key* sk*, and predicate vector* $\overrightarrow{v}$*. It outputs a corresponding secret key* $\mathsf{sk}_{\overrightarrow{v}}$.
- Enc *takes as input the master public key* pk*, plaintext* m *in some associated plaintext space,* msg*, and attribute vector* $\overrightarrow{x}$*. It returns ciphertext* c.
- Dec *takes as input the master public key* pk*, secret key* $\mathsf{sk}_{\overrightarrow{v}}$ *and ciphertext* c*. It outputs either plaintext* m *or the distinguished symbol* $\perp$.

A PE scheme should have the following correctness property: for all $f_{\overrightarrow{v}} \in \mathcal{F}$ and $\overrightarrow{x} \in \Sigma$, for correctly generated pk, $\mathsf{sk}_{\overrightarrow{v}}$ and $c \xleftarrow{\mathsf{R}} \mathsf{Enc}(\mathsf{pk}, m, \overrightarrow{x})$, it holds that $m = \mathsf{Dec}(\mathsf{pk}, \mathsf{sk}_{\overrightarrow{v}}, c)$ if $f_{\overrightarrow{v}}(\overrightarrow{x}) = 1$. Otherwise, it holds with negligible probability.

Definition 17 *An inner-product predicate encryption scheme is* adaptively attribute-hiding (AH) against chosen plaintext attacks *if for all probabilistic polynomial-time adversaries* $\mathcal{A}$*, the advantage of* $\mathcal{A}$ *in the following experiment is negligible in the security parameter.*

1. Setup *is run to generate keys* pk *and* sk*, and* pk *is given to* $\mathcal{A}$.
2. $\mathcal{A}$ *may adaptively make a polynomial number of key queries for predicate vectors,* $\overrightarrow{v}$*. In response,* $\mathcal{A}$ *is given the corresponding key* $\mathsf{sk}_{\overrightarrow{v}} \xleftarrow{\mathsf{R}} \mathsf{KeyGen}(\mathsf{sk}, \overrightarrow{v})$.
3. $\mathcal{A}$ *outputs challenge attribute vector* $(\overrightarrow{x}^{(0)}, \overrightarrow{x}^{(1)})$ *and challenge plaintexts* $(m^{(0)}, m^{(1)})$*, subject to the restriction that* $\overrightarrow{v} \cdot \overrightarrow{x}^{(0)} \neq 0$ *and* $\overrightarrow{v} \cdot \overrightarrow{x}^{(1)} \neq 0$ *for all the key queried predicate vectors,* $\overrightarrow{v}$.
4. *A random bit* b *is chosen.* $\mathcal{A}$ *is given* $c^{(b)} \xleftarrow{\mathsf{R}} \mathsf{Enc}(\mathsf{pk}, m^{(b)}, \overrightarrow{x}^{(b)})$.
5. *The adversary may continue to issue key queries for additional predicate vectors,* $\overrightarrow{v}$*, subject to the restriction that* $\overrightarrow{v} \cdot \overrightarrow{x}^{(0)} \neq 0$ *and* $\overrightarrow{v} \cdot \overrightarrow{x}^{(1)} \neq 0$*.* $\mathcal{A}$ *is given the corresponding key* $\mathsf{sk}_{\overrightarrow{v}} \xleftarrow{\mathsf{R}} \mathsf{KeyGen}(\mathsf{sk}, \overrightarrow{v})$.
6. $\mathcal{A}$ *outputs a bit* b'*, and succeeds if* $b' = b$.

We define the advantage of $\mathcal{A}$ *as the quantity* $\mathsf{Adv}^{\mathsf{PE,AH}}_{\mathcal{A}}(\lambda) := \mathsf{Pr}\left[b' = b\right] - 1/2$.

Remark: In Definition 17, adversary $\mathcal{A}$ is not allowed to ask a key query for $\overrightarrow{v}$ such that $\overrightarrow{v} \cdot \overrightarrow{x}^{(b)} = 0$ for some $b \in \{0, 1\}$, while in the security definition in [29], such a key query is allowed provided that $m^{(0)} = m^{(1)}$ and $\overrightarrow{v} \cdot \overrightarrow{x}^{(b)} = 0$ for all $b \in \{0, 1\}$.

3.5 The Proposed PE Scheme

3.5.1 Construction

$$
\begin{aligned}
&\mathsf{Setup}(1^\lambda, n):\ (\mathsf{param}_{\mathbb{V}}, \mathbb{B}, \mathbb{B}^*) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^\lambda, 2n+3),\\
&\quad \widehat{\mathbb{B}} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n, \boldsymbol{b}_{2n+1}, \boldsymbol{b}_{2n+3}),\ \ \mathsf{sk} := \mathbb{B}^*,\ \ \mathsf{pk} := (1^\lambda, \mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}),\\
&\quad \text{return}\ \ \mathsf{sk},\ \mathsf{pk}.\\
&\mathsf{KeyGen}(\mathsf{sk}, \overrightarrow{v} := (v_1, \ldots, v_n)):\ \sigma, \eta \xleftarrow{\mathsf{U}} \mathbb{F}_q,\\
&\quad \boldsymbol{k}^* := \sigma(\textstyle\sum_{i=1}^n v_i \boldsymbol{b}_i^*) + \boldsymbol{b}_{2n+1}^* + \eta \boldsymbol{b}_{2n+2}^*,\\
&\quad \text{return}\ \ \mathsf{sk}_{\overrightarrow{v}} := \boldsymbol{k}^*.\\
&\mathsf{Enc}(\mathsf{pk}, m \in \mathbb{G}_T, \overrightarrow{x} := (x_1, \ldots, x_n)):\ \delta_1, \delta_2, \zeta \xleftarrow{\mathsf{U}} \mathbb{F}_q,\\
&\quad \boldsymbol{c}_1 := \delta_1(\textstyle\sum_{i=1}^n x_i \boldsymbol{b}_i) + \zeta \boldsymbol{b}_{2n+1} + \delta_2 \boldsymbol{b}_{2n+3},\quad c_2 := g_T^\zeta m,\\
&\quad \text{return}\ (\boldsymbol{c}_1, c_2).\\
&\mathsf{Dec}(\mathsf{pk}, \boldsymbol{k}^*, (\boldsymbol{c}_1, c_2)):\ m' := c_2 / e(\boldsymbol{c}_1, \boldsymbol{k}^*),\\
&\quad \text{return}\ \ m'.
\end{aligned}
$$

[Correctness] $\boldsymbol{k}^*$ and $\boldsymbol{c}_1$ can be expressed by $\boldsymbol{k}^* = (\sigma\overrightarrow{v}, 0, \ldots, 0, 1, \eta, 0)_{\mathbb{B}^*}$, and $\boldsymbol{c}_1 = (\delta_1\overrightarrow{x}, 0, \ldots, 0, \zeta, 0, \delta_2)_{\mathbb{B}}$. Hence, $e(\boldsymbol{c}_1, \boldsymbol{k}^*) = g_T^{(\delta_1\overrightarrow{x}, 0, \ldots, 0, \zeta, 0, \delta_2)\cdot(\sigma\overrightarrow{v}, 0, \ldots, 0, 1, \eta, 0)} = g_T^{\delta_1\sigma(\overrightarrow{x}\cdot\overrightarrow{v})+\zeta}$, i.e., $e(\boldsymbol{c}_1, \boldsymbol{k}^*) = g_T^\zeta$ if $\overrightarrow{x}\cdot\overrightarrow{v} = 0$.

3.5.2 Security

Theorem 18 *The proposed PE scheme is adaptively attribute-hiding against chosen plaintext attacks under the n-eDDH assumption. For any adversary $\mathcal{A}$, there exist probabilistic machines $\mathcal{C}_k$ ($k = 0, \ldots, \nu$), whose running times are essentially the same as that of $\mathcal{A}$, such that for any security parameter λ,*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{PE,AH}}(\lambda) \le \sum_{k=0}^{\nu} \mathsf{Adv}_{\mathcal{C}_k}^{n\text{-eDDH}}(\lambda) + \frac{\nu}{q},$$

where ν is the maximum number of adversary $\mathcal{A}$'s key queries.

We will show Lemmas 20, 22, and 23 for the proof of Theorem 18.

Definition 19 *Problem 1 is to guess $\beta \in \{0,1\}$, given $(\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \{\boldsymbol{e}_{\beta,i}\}_{i=1,..,n}) \xleftarrow{\mathsf{R}} \mathcal{G}_\beta^{\mathsf{P1}}(1^\lambda, n)$, where*

$$
\begin{aligned}
&\mathcal{G}_\beta^{\mathsf{P1}}(1^\lambda, n):\ (\mathsf{param}_{\mathbb{V}}, \mathbb{B}, \mathbb{B}^*) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^\lambda, 2n+3),\\
&\quad \widehat{\mathbb{B}} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n, \boldsymbol{b}_{2n+1}, \boldsymbol{b}_{2n+3}),\quad \widehat{\mathbb{B}}^* := (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*, \boldsymbol{b}_{2n+1}^*, \boldsymbol{b}_{2n+2}^*),\\
&\quad \delta_1, \delta_{2,i} \xleftarrow{\mathsf{U}} \mathbb{F}_q,\quad \rho \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times,\quad (u_{i,j}) \xleftarrow{\mathsf{U}} GL(n, \mathbb{F}_q)\ \ \text{for}\ i, j = 1, \ldots, n,\\
&\quad \text{for}\ i = 1, \ldots, n,\\
&\qquad \boldsymbol{e}_{0,i} := \delta_1 \boldsymbol{b}_i + \delta_{2,i} \boldsymbol{b}_{2n+3},\\
&\qquad \boldsymbol{e}_{1,i} := \delta_1 \boldsymbol{b}_i + \rho \textstyle\sum_{j=1}^n u_{i,j} \boldsymbol{b}_{n+j} + \delta_{2,i} \boldsymbol{b}_{2n+3},\\
&\quad \text{return}\ \ (\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \{\boldsymbol{e}_{\beta,i}\}_{i=1,\ldots,n}),
\end{aligned}
$$

for $\beta \xleftarrow{\mathsf{U}} \{0,1\}$. For a probabilistic machine $\mathcal{B}$, we define the advantage of $\mathcal{B}$ for Problem 1 as:

$$\mathsf{Adv}_{\mathcal{B}}^{\mathsf{P1}}(\lambda) := \left| \mathsf{Pr}\left[\mathcal{B}(1^\lambda, \varrho) \to 1 \,\middle|\, \varrho \xleftarrow{\mathsf{R}} \mathcal{G}_0^{\mathsf{P1}}(1^\lambda, n)\right] - \mathsf{Pr}\left[\mathcal{B}(1^\lambda, \varrho) \to 1 \,\middle|\, \varrho \xleftarrow{\mathsf{R}} \mathcal{G}_1^{\mathsf{P1}}(1^\lambda, n)\right] \right|.$$

Lemma 20 *For any adversary $\mathcal{B}$, there is a probabilistic machine $\mathcal{C}$, whose running time is essentially the same as that of $\mathcal{B}$, such that for any security parameter λ, $\mathsf{Adv}^{n\text{-}\mathsf{eDDH}}_{\mathcal{C}}(\lambda) = \mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}}(\lambda)$.*

The proof of Lemma 20 is given in Appendix B.3.2.

Definition 21 *Problem 2 is to guess $\beta \in \{0,1\}$, given $(\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \{\boldsymbol{h}^*_{\beta,i},\ \boldsymbol{e}_i\}_{i=1,\ldots,n}) \xleftarrow{\mathsf{R}} \mathcal{G}^{\mathsf{P2}}_{\beta}(1^\lambda, n)$, where*

$$
\begin{aligned}
&\mathcal{G}^{\mathsf{P2}}_{\beta}(1^\lambda, n):\ (\mathsf{param}_{\mathbb{V}}, \mathbb{B}, \mathbb{B}^*) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^\lambda, 2n+3),\\
&\quad \widehat{\mathbb{B}} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n, \boldsymbol{b}_{2n+1}, \boldsymbol{b}_{2n+3}), \quad \widehat{\mathbb{B}}^* := (\boldsymbol{b}^*_1, \ldots, \boldsymbol{b}^*_{2n+2}),\\
&\quad \omega, \gamma_i, \delta \xleftarrow{\mathsf{U}} \mathbb{F}_q,\ \rho, \tau \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times,\\
&\quad (u_{i,j}) \xleftarrow{\mathsf{U}} GL(n, \mathbb{F}_q), \quad (z_{i,j}) := ((u_{i,j})^{-1})^{\mathrm{T}} \ \text{ for } i,j = 1,\ldots,n,\\
&\quad \text{for } i = 1,\ldots,n,\\
&\qquad \boldsymbol{h}^*_{0,i} := \omega \boldsymbol{b}^*_i + \gamma_i \boldsymbol{b}^*_{2n+2},\\
&\qquad \boldsymbol{h}^*_{1,i} := \omega \boldsymbol{b}^*_i + \tau \textstyle\sum_{j=1}^n z_{i,j} \boldsymbol{b}^*_{n+j} + \gamma_i \boldsymbol{b}^*_{2n+2},\\
&\qquad \boldsymbol{e}_i := \delta \boldsymbol{b}_i + \rho \textstyle\sum_{j=1}^n u_{i,j} \boldsymbol{b}_{n+j},\\
&\quad \text{return } \ (\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \{\boldsymbol{h}^*_{\beta,i}, \boldsymbol{e}_i\}_{i=1,\ldots,n}),
\end{aligned}
$$

for $\beta \xleftarrow{\mathsf{U}} \{0,1\}$. For a probabilistic machine $\mathcal{B}$, the advantage of $\mathcal{B}$ for Problem 2, $\mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}}(\lambda)$, is similarly defined as in Definition 19.

Lemma 22 *For any adversary $\mathcal{B}$, there is a probabilistic machine $\mathcal{C}$, whose running time is essentially the same as that of $\mathcal{B}$, such that for any security parameter λ, $\mathsf{Adv}^{n\text{-}\mathsf{eDDH}}_{\mathcal{C}}(\lambda) = \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}}(\lambda)$.*

The proof of Lemma 22 is given in Appendix B.3.3.

Lemma 23 *Let $C := \{(\overrightarrow{x}, \overrightarrow{v}) | \overrightarrow{x} \cdot \overrightarrow{v} \neq 0\} \subset V \times V^*$ where V is n-dimensional vector space $\mathbb{F}_q^n$, and V^* its dual. For all $(\overrightarrow{x}, \overrightarrow{v}) \in C$, for all $(\overrightarrow{r}, \overrightarrow{w}) \in C$,*

$$
\Pr_{\substack{Z \xleftarrow{\mathsf{U}} GL(n,\mathbb{F}_q),\\ \rho,\tau \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times}} [\overrightarrow{x}\,(\rho U) = \overrightarrow{r} \ \wedge \ \overrightarrow{v}\,(\tau Z) = \overrightarrow{w}] = \frac{1}{s},
$$

where $U := (Z^{-1})^{\mathrm{T}}$ and $s := \sharp C$ $(= (q^n - 1)(q^n - q^{n-1}))$.

The proof of Lemma 23 is given in Appendix B.4.

Proof Outline of Theorem 18: To prove the security, we employ Game 0 (original adaptive-security game) through Game 3. Roughly speaking, the (normal) target ciphertext is changed to a *semi-functional* ciphertext in Game 1 (or Game 2-0), the k-th secret key replied to the adversary is changed to a *semi-functional* key in Game 2-k ($k = 1, \ldots, \nu$), and the (semi-functional) target ciphertext is changed to perfectly *randomized* key in Game 3, whose advantage is 0.

A *normal* secret key $\boldsymbol{k}^{*\,\mathsf{norm}}_{\overrightarrow{v}}$ (with predicate vector $\overrightarrow{v}$) is a correct form of the secret key of the proposed PE scheme, i.e., $\boldsymbol{k}^{*\,\mathsf{norm}}_{\overrightarrow{v}} := \sigma(\sum_{i=1}^n v_i \boldsymbol{b}^*_i) + \boldsymbol{b}^*_{2n+1} + \eta \boldsymbol{b}^*_{2n+2} = (\sigma\overrightarrow{v}, \overrightarrow{0}_n, 1, \eta, 0)_{\mathbb{B}^*}$,

where $\overrightarrow{0}_n := (\overbrace{0, \cdots, 0}^{n})$. Similarly, a *normal* ciphertext (with attribute $\overrightarrow{x}$) is $(\boldsymbol{c}_{\overrightarrow{x}}^{\mathsf{norm}}, c_2)$ with $\boldsymbol{c}_{\overrightarrow{x}}^{\mathsf{norm}} := \delta_1(\sum_{i=1}^{n} x_i \boldsymbol{b}_i) + \zeta \boldsymbol{b}_{2n+1} + \delta_2 \boldsymbol{b}_{2n+3} = (\delta_1 \overrightarrow{x}, \overrightarrow{0}_n, \zeta, 0, \delta_2)_{\mathbb{B}}$. (Hereafter we will ignore c_2 since c_2 is always correctly generated.) A *semi-functional* secret key is $\boldsymbol{k}_{\overrightarrow{v}}^{*\,\mathsf{semi}} := (\sigma \overrightarrow{v}, \overrightarrow{r}, 1, \eta, 0)_{\mathbb{B}^*}$ and a *semi-functional* ciphertext is $\boldsymbol{c}_{\overrightarrow{x}}^{\mathsf{semi}} := (\delta_1 \overrightarrow{x}, \overrightarrow{s}, \zeta, 0, \delta_2)_{\mathbb{B}}$, where $\overrightarrow{r}, \overrightarrow{s} \xleftarrow{\mathsf{U}} \mathbb{F}_q^n$. If $\overrightarrow{x} \cdot \overrightarrow{v} = 0$, then $e(\boldsymbol{c}_{\overrightarrow{x}}^{\mathsf{norm}}, \boldsymbol{k}_{\overrightarrow{v}}^{*\,\mathsf{norm}}) = e(\boldsymbol{c}_{\overrightarrow{x}}^{\mathsf{norm}}, \boldsymbol{k}_{\overrightarrow{v}}^{*\,\mathsf{semi}}) = e(\boldsymbol{c}_{\overrightarrow{x}}^{\mathsf{semi}}, \boldsymbol{k}_{\overrightarrow{v}}^{*\,\mathsf{norm}}) = g_T^{\zeta}$, which leads to correct decryption. In contrast, $e(\boldsymbol{c}_{\overrightarrow{x}}^{\mathsf{semi}}, \boldsymbol{k}_{\overrightarrow{v}}^{*\,\mathsf{semi}}) = g_T^{\overrightarrow{s} \cdot \overrightarrow{r} + \zeta}$, which is uniformly and independently distributed over $\mathbb{F}_q$ since $\overrightarrow{r}, \overrightarrow{s} \xleftarrow{\mathsf{U}} \mathbb{F}_q^n$, (i.e., leads to random decryption).

To prove that the advantage gap between Games 0 and 1 is bounded by the advantage of Problem 1 (to guess $\beta \in \{0,1\}$), we construct a simulator of the challenger of Game 0 (or 1) (against an adversary $\mathcal{A}$) by using an instance with $\beta \xleftarrow{\mathsf{U}} \{0,1\}$ of Problem 1. We then show that the distribution of the secret keys and target ciphertext replied by the simulator is equivalent to those of Game 0 when $\beta = 0$ and Game 1 when $\beta = 1$. That is, the advantage of Problem 1 is equivalent to the advantage gap between Games 0 and 1 (Lemma 24). The advantage of Problem 1 is proven to be equivalent to that of the n-eDDH assumption (Lemma 20).

The advantage gap between Games 2-$(k-1)$ and 2-k is similarly shown to be bounded by the advantage of Problem 2 (i.e., of the n-eDDH assumption) $+1/q$ (Lemmas 22 and 25).

Problem 2 is based on our key trick (explained in Section 3.1.2). Here, we introduce *special form of semi-functional* keys and ciphertexts such that $\boldsymbol{k}_{\overrightarrow{v}}^{*\,\mathsf{spec.semi}} := (\sigma \overrightarrow{v}, (\tau \overrightarrow{v} Z), 1, \eta, 0)_{\mathbb{B}^*}$, and $\boldsymbol{c}_{\overrightarrow{x}}^{\mathsf{spec.semi}} := (\delta \overrightarrow{x}, (\rho \overrightarrow{x} U), \zeta, 0, \delta_2)_{\mathbb{B}}$, where Z is a random regular $(n \times n)$-matrix, $U := (Z^{-1})^{\mathrm{T}}$, and $\tau, \rho \xleftarrow{\mathsf{U}} \mathbb{F}_q$.

$\boldsymbol{k}_{\overrightarrow{v}}^{*\,\mathsf{spec.semi}}$ can decrypt $\boldsymbol{c}_{\overrightarrow{x}}^{\mathsf{spec.semi}}$ for *all* vectors $\overrightarrow{x}$ with $\overrightarrow{v} \cdot \overrightarrow{x} = 0$, since $(\tau \overrightarrow{v} Z) \cdot (\rho \overrightarrow{x} U) = \tau\rho(\overrightarrow{v} \cdot \overrightarrow{x})$, i.e., $e(\boldsymbol{c}_{\overrightarrow{x}}^{\mathsf{spec.semi}}, \boldsymbol{k}_{\overrightarrow{v}}^{*\,\mathsf{spec.semi}}) = g^{(\delta_1\sigma + \tau\rho)(\overrightarrow{v} \cdot \overrightarrow{x}) + \zeta}$. In addition, $(\tau \overrightarrow{v} Z)$ and $(\rho \overrightarrow{x} U)$ are uniformly and pairwise-independently distributed (i.e., equivalently distributed to $(\overrightarrow{r}, \overrightarrow{s}) \xleftarrow{\mathsf{U}} (\mathbb{F}_q^n)^2 \backslash \{(\overrightarrow{r}, \overrightarrow{s}) \mid \overrightarrow{r} \cdot \overrightarrow{s} = 0\}$), when $\overrightarrow{v} \cdot \overrightarrow{x} \neq 0$ (Lemma 23). Therefore, the joint distribution of $\boldsymbol{k}_{\overrightarrow{v}}^{*\,\mathsf{spec.semi}}$ and $\boldsymbol{c}_{\overrightarrow{x}}^{\mathsf{spec.semi}}$ is equivalent to that of an independent pair of $\boldsymbol{k}_{\overrightarrow{v}}^{*\,\mathsf{semi}}$ and $\boldsymbol{c}_{\overrightarrow{x}}^{\mathsf{semi}}$ (except with probability $1/q$), when $\overrightarrow{v} \cdot \overrightarrow{x} \neq 0$.

Finally we show that Game 2-ν can be conceptually changed to Game 3 by using the fact that n elements of $\mathbb{B}$, $(\boldsymbol{b}_{n+1}, \ldots, \boldsymbol{b}_{2n})$, are secret to the adversary (Lemma 26).

Proof of Theorem 18: To prove Theorem 18, we consider the following $(\nu + 3)$ games.

Game 0 Original game.

Game 1 Same as Game 0 except that the target ciphertext $(\boldsymbol{c}_1, c_2)$ for challenge plaintexts $(m^{(0)}, m^{(1)})$ and challenge attributes $(\overrightarrow{x}^{(0)}, \overrightarrow{x}^{(1)})$ is

$$\boldsymbol{c}_1 := \delta_1(\textstyle\sum_{i=1}^{n} x_i^{(b)} \boldsymbol{b}_i) + \sum_{i=1}^{n} w_i \boldsymbol{b}_{n+i} + \zeta \boldsymbol{b}_{2n+1} + \delta_2 \boldsymbol{b}_{2n+3}, \quad c_2 := g_T^{\zeta} m^{(b)},$$

where $\delta_1, \delta_2, \zeta \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $b \xleftarrow{\mathsf{U}} \{0,1\}$, $(x_1^{(b)}, \ldots, x_n^{(b)}) := \overrightarrow{x}^{(b)}$, and $(w_1, \ldots, w_n) \xleftarrow{\mathsf{U}} \mathbb{F}_q^n \setminus \{\overrightarrow{0}\}$.

Game 2-k ($k = 1, \ldots, \nu$) Game 2-0 is Game 1. Game 2-k is the same as Game 2-$(k-1)$ except the reply to the k-th key query for $\overrightarrow{v} := (v_1, \ldots, v_n)$ is:

$$\boldsymbol{k}^* := \sigma(\textstyle\sum_{i=1}^{n} v_i \boldsymbol{b}_i^*) + \sum_{i=1}^{n} r_i \boldsymbol{b}_{n+i}^* + \boldsymbol{b}_{2n+1}^* + \eta \boldsymbol{b}_{2n+2}^*,$$

where $\sigma, \eta \xleftarrow{\mathsf{U}} \mathbb{F}_q$ and $\overrightarrow{r} := (r_1, \ldots, r_n) \xleftarrow{\mathsf{U}} \mathbb{F}_q^n$.

Game 3 Same as Game 2-ν except that the target ciphertext $(\boldsymbol{c}_1, c_2)$ for challenge plaintexts $(m^{(0)}, m^{(1)})$ and challenge attributes $(\overrightarrow{x}^{(0)}, \overrightarrow{x}^{(1)})$ is

$$\boldsymbol{c}_1 := \sum_{i=1}^{n} x'_i \boldsymbol{b}_i + \sum_{i=1}^{n} w_i \boldsymbol{b}_{n+i} + \zeta' \boldsymbol{b}_{2n+1} + \delta_2 \boldsymbol{b}_{2n+3}, \quad c_2 := g_T^{\zeta} m^{(b)},$$

where $x'_1, \ldots, x'_n, \delta_2, \zeta, \zeta' \xleftarrow{\mathsf{U}} \mathbb{F}_q, b \xleftarrow{\mathsf{U}} \{0,1\}$, and $(w_1, \ldots, w_n) \xleftarrow{\mathsf{U}} \mathbb{F}_q^n \setminus \{\overrightarrow{0}\}$. In particular, we note that $(x'_1, \ldots, x'_n)$ and ζ' are chosen uniformly and independently from $\overrightarrow{x}^{(0)}, \overrightarrow{x}^{(1)}$ and ζ.

Let $\mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda)$ be $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{PE,AH}}(\lambda)$ in Game 0, and $\mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda), \mathsf{Adv}_{\mathcal{A}}^{(2\text{-}k)}(\lambda), \mathsf{Adv}_{\mathcal{A}}^{(3)}(\lambda)$ be the advantage of $\mathcal{A}$ in Game $1, 2\text{-}k, 3$, respectively. It is clear that $\mathsf{Adv}_{\mathcal{A}}^{(3)}(\lambda) = 0$ by Lemma 27.

We will use three lemmas (Lemmas 24, 25, 26) that evaluate the gaps between pairs of $\mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda), \mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda), \mathsf{Adv}_{\mathcal{A}}^{(2\text{-}k)}(\lambda)$ $(k = 1, \ldots, \nu), \mathsf{Adv}_{\mathcal{A}}^{(3)}(\lambda)$. From these lemmas, we obtain $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{PE,AH}}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda) \leq \left|\mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda)\right| + \sum_{k=1}^{\nu} \left|\mathsf{Adv}_{\mathcal{A}}^{(2\text{-}(k-1))}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(2\text{-}k)}(\lambda)\right| + \left|\mathsf{Adv}_{\mathcal{A}}^{(2\text{-}\nu)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(3)}(\lambda)\right| + \mathsf{Adv}_{\mathcal{A}}^{(3)}(\lambda) \leq \mathsf{Adv}_{\mathcal{B}_0}^{\mathsf{P1}}(\lambda) + \sum_{k=1}^{\nu} \mathsf{Adv}_{\mathcal{B}_k}^{\mathsf{P2}}(\lambda) + \frac{\nu}{q}$. From Lemmas 20 and 22, there exist probabilistic machines $\mathcal{C}_k$ $(k = 0, \ldots, \nu)$, whose running times are essentially the same as those of $\mathcal{B}_k$, respectively, such that $\mathsf{Adv}_{\mathcal{C}_0}^{n\text{-eDDH}}(\lambda) = \mathsf{Adv}_{\mathcal{B}_0}^{\mathsf{P2}}(\lambda)$ and $\mathsf{Adv}_{\mathcal{C}_k}^{n\text{-eDDH}}(\lambda) = \mathsf{Adv}_{\mathcal{B}_k}^{\mathsf{P2}}(\lambda)$ $(k = 1, \ldots, \nu)$. Hence, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{PE,AH}}(\lambda) \leq \mathsf{Adv}_{\mathcal{B}_0}^{\mathsf{P1}}(\lambda) + \sum_{k=1}^{\nu} \mathsf{Adv}_{\mathcal{B}_k}^{\mathsf{P2}}(\lambda) + \frac{\nu}{q} \leq \sum_{k=0}^{\nu} \mathsf{Adv}_{\mathcal{C}_k}^{n\text{-eDDH}}(\lambda) + \frac{\nu}{q}$. This completes the proof of Theorem 18. □

Lemma 24 *For any adversary $\mathcal{A}$, there exists a probabilistic machine $\mathcal{B}_0$, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $|\mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda)| = \mathsf{Adv}_{\mathcal{B}_0}^{\mathsf{P1}}(\lambda)$.*

Proof. In order to prove Lemma 24, we construct a probabilistic machine $\mathcal{B}_0$ against Problem 1 by using any adversary $\mathcal{A}$ in a security game (Game 0 or 1) as a black box as follows:

1. $\mathcal{B}_0$ is given Problem 1 instance $(\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \{\boldsymbol{e}_{\beta,i}\}_{i=1,\ldots,n})$.
2. $\mathcal{B}_0$ plays a role of the challenger in the security game against adversary $\mathcal{A}$.
3. At the first step of the game, $\mathcal{B}_0$ returns $\mathsf{pk} := (1^{\lambda}, \mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}})$ to $\mathcal{A}$.
4. When a key query is issued, $\mathcal{B}_0$ answers a correct secret key computed by using $\widehat{\mathbb{B}}^*$, i.e., normal key.
5. When $\mathcal{B}_0$ gets challenge plaintexts $(m^{(0)}, m^{(1)})$ and challenge attributes $(\overrightarrow{x}^{(0)}, \overrightarrow{x}^{(1)})$ (from $\mathcal{A}$), $\mathcal{B}_0$ calculates and returns $(\boldsymbol{c}_1, c_2)$ such that $\boldsymbol{c}_1 := \sum_{i=1}^{n} x_i^{(b)} \boldsymbol{e}_{\beta,i} + \zeta \boldsymbol{b}_{n+1}$ and $c_2 := g_T^{\zeta} m^{(b)}$ where $\boldsymbol{e}_{\beta,i}$ are from the Problem 1 instance, $\zeta \xleftarrow{\mathsf{U}} \mathbb{F}_q$ and $b \xleftarrow{\mathsf{U}} \{0,1\}$.
6. After the challenge encryption query, KeyGen oracle simulation for a key query is executed in the same manner as step 4.
7. $\mathcal{A}$ outputs bit b'. If $b = b'$, $\mathcal{B}_0$ outputs $\beta' := 1$. Otherwise, $\mathcal{B}_0$ outputs $\beta' := 0$.

Claim 1 *If $\beta = 0$, the distribution of $(\boldsymbol{c}_1, c_2)$ generated in step 5 is the same as that in Game 0. If $\beta = 1$, the distribution of $(\boldsymbol{c}_1, c_2)$ generated in step 5 is the same as that in Game 1.*

Proof. If $\beta = 0$, $\boldsymbol{c}_1 = \delta_1 \sum_{i=1}^n x_i \boldsymbol{b}_i + \zeta \boldsymbol{b}_{2n+1} + (\sum_{i=1}^n x_i \delta_{2,i})\, \boldsymbol{b}_{2n+3}$ and $c_2 := g_T^{\zeta} m^{(b)}$. This is the target ciphertext in Game 0. If $\beta = 1$,

$$\boldsymbol{c}_1 = \delta_1 \textstyle\sum_{i=1}^n x_i \boldsymbol{b}_i + \rho \sum_{i=1}^n (\overrightarrow{x} \cdot \overrightarrow{u}_i)\, \boldsymbol{b}_{n+i} + \zeta \boldsymbol{b}_{2n+1} + \left(\overrightarrow{x} \cdot \overrightarrow{\delta}_2\right) \boldsymbol{b}_{2n+3},$$

where $\overrightarrow{u}_i := (u_{1,i}, \ldots, u_{n,i})$, $\overrightarrow{\delta}_2 := (\delta_{2,1}, \ldots, \delta_{2,n})$, and $c_2 = g_T^{\zeta} m^{(b)}$. Because $(\rho \overrightarrow{x} \cdot \overrightarrow{u}_1, \ldots, \rho \overrightarrow{x} \cdot \overrightarrow{u}_n) \in \mathbb{F}_q^n \setminus \{\overrightarrow{0}\}$ and $\overrightarrow{x} \cdot \overrightarrow{\delta}_2 \in \mathbb{F}_q$ are independently uniform, this is the target ciphertext in Game 1. □

From Claim 1, when $\beta = 0$, the advantage of $\mathcal{A}$ in the above game is equal to that in Game 0, i.e., $\mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda)$, and is also equal to $\mathsf{Pr}_0 := \mathsf{Pr}\left[\mathcal{B}_0(1^\lambda, \varrho) \to 1 \;\middle|\; \varrho \xleftarrow{\mathsf{R}} \mathcal{G}_0^{\mathsf{P1}}(1^\lambda, n)\right]$. Similarly, when $\beta = 1$, we see that the advantage of $\mathcal{A}$ in the above game is equal to $\mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda)$, and is also equal to $\mathsf{Pr}_1 := \mathsf{Pr}\left[\mathcal{B}_0(1^\lambda, \varrho) \to 1 \;\middle|\; \varrho \xleftarrow{\mathsf{R}} \mathcal{G}_1^{\mathsf{P1}}(1^\lambda, n)\right]$. Therefore, $|\mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda)| = |\mathsf{Pr}_0 - \mathsf{Pr}_1| = \mathsf{Adv}_{\mathcal{B}_0}^{\mathsf{P1}}(\lambda)$. This completes the proof of Lemma 24. □

Lemma 25 *For any adversary $\mathcal{A}$, there exists a probabilistic machine $\mathcal{B}_k$, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $|\mathsf{Adv}_{\mathcal{A}}^{(2\text{-}(k-1))}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(2\text{-}k)}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}_k}^{\mathsf{P2}}(\lambda) + \frac{1}{q}$.*

Proof. In order to prove Lemma 25, we construct a probabilistic machine $\mathcal{B}_k$ against Problem 2 by using any adversary $\mathcal{A}$ in a security game (Game 2-$(k-1)$ or 2-k) as a black box as follows:

1. $\mathcal{B}_k$ is given Problem 2 instance $(\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \{\boldsymbol{h}^*_{\beta,i}, \boldsymbol{e}_i\}_{i=1,\ldots,n})$.
2. $\mathcal{B}_k$ plays a role of the challenger in the security game against adversary $\mathcal{A}$.
3. At the first step of the game, $\mathcal{B}_k$ returns $\mathsf{pk} := (1^\lambda, \mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}})$ to $\mathcal{A}$.
4. When the s-th key query is issued for a predicate $\overrightarrow{v} := (v_1, \ldots, v_n)$, $\mathcal{B}_k$ answers as follows:
   (a) When $1 \leq s \leq k-1$, $\mathcal{B}_k$ calculates and answers (by using $\widehat{\mathbb{B}}^*$)
   $$\boldsymbol{k}^* := \sigma(\textstyle\sum_{i=1}^n v_i \boldsymbol{b}^*_i) + \sum_{i=1}^n r_i \boldsymbol{b}^*_{n+i} + \boldsymbol{b}^*_{2n+1} + \eta \boldsymbol{b}^*_{2n+2},$$
   where $\sigma, r_1, \ldots, r_n, \eta \xleftarrow{\mathsf{U}} \mathbb{F}_q$ (i.e., semi-functional key).
   (b) When $s = k$, $\mathcal{B}_k$ calculates and answers $\boldsymbol{k}^*$ as follows:
   $$\boldsymbol{k}^* := \textstyle\sum_{i=1}^n v_i \boldsymbol{h}^*_{\beta,i} + \boldsymbol{b}^*_{2n+1}.$$
   (c) When $s \geq k+1$, $\mathcal{B}_k$ answers a correct secret key computed by using $\widehat{\mathbb{B}}^*$, i.e., normal key.
5. When $\mathcal{B}_k$ gets challenge plaintexts $(m^{(0)}, m^{(1)})$ and challenge attributes $(\overrightarrow{x}^{(0)}, \overrightarrow{x}^{(1)})$ (from $\mathcal{A}$), $\mathcal{B}_k$ calculates and returns $(\boldsymbol{c}_1, c_2)$ such that
   $$\boldsymbol{c}_1 := \textstyle\sum_{i=1}^n x_i^{(b)} \boldsymbol{e}_i + \zeta \boldsymbol{b}_{2n+1} + \delta_2 \boldsymbol{b}_{2n+3}, \quad c_2 := g_T^{\zeta} m^{(b)},$$
   where $\boldsymbol{e}_i$ are from the Problem 2 instance, $\zeta, \delta_2 \xleftarrow{\mathsf{U}} \mathbb{F}_q$ and $b \xleftarrow{\mathsf{U}} \{0,1\}$.

6. After the challenge encryption query, KeyGen oracle simulation for a key query is executed as in step 4.

7. $\mathcal{A}$ outputs bit b'. If $b = b'$, $\mathcal{B}_k$ outputs $\beta' := 1$. Otherwise, $\mathcal{B}_k$ outputs $\beta' := 0$.

Claim 2 *The pair of secret key $\overrightarrow{\boldsymbol{k}}^*$ generated in case (b) of step 4 or 6 and ciphertext $\boldsymbol{c}_1$ generated in step 5 has the same distribution as that in Game 2-$(k-1)$ (resp. Game 2-k) when $\beta = 0$ (resp. $\beta = 1$) except with probability $\frac{1}{q}$.*

Proof. We consider the joint distribution of $\boldsymbol{c}_1$ and $\boldsymbol{k}^*$.

Ciphertext $\boldsymbol{c}_1$ generated in step 5 is

$$\boldsymbol{c}_1 := \delta(\textstyle\sum_{i=1}^n x_i \boldsymbol{b}_i) + \rho(\sum_{i=1}^n (\overrightarrow{x} \cdot \overrightarrow{u}_i)\boldsymbol{b}_i) + \zeta \boldsymbol{b}_{2n+1} + \delta_2 \boldsymbol{b}_{2n+3},$$

where $\overrightarrow{u}_i := (u_{1,i}, \ldots, u_{n,i})$. Then, $\delta, \zeta, \delta_2 \in \mathbb{F}_q$ and $(\rho(\overrightarrow{x} \cdot \overrightarrow{u}_1), \ldots, \rho(\overrightarrow{x} \cdot \overrightarrow{u}_n)) \in \mathbb{F}_q^n \setminus \{\overrightarrow{0}\}$ are independently uniform.

When $\beta = 0$, secret key $\boldsymbol{k}^*$ generated in case (b) of step 4 or 6 is

$$\boldsymbol{k}^* = \omega \textstyle\sum_{i=1}^n v_i \boldsymbol{b}_i^* + \boldsymbol{b}_{2n+1}^* + (\overrightarrow{v} \cdot \overrightarrow{\gamma})\boldsymbol{b}_{2n+2}^*,$$

where $\overrightarrow{\gamma} := (\gamma_1, \ldots, \gamma_n)$. Then, $\omega, \overrightarrow{v} \cdot \overrightarrow{\gamma}$ and the above coefficients in $\boldsymbol{c}_1$, i.e., $\delta, \zeta, \delta_2, \rho(\overrightarrow{x} \cdot \overrightarrow{u}_1), \ldots, \rho(\overrightarrow{x} \cdot \overrightarrow{u}_n)$, are independently uniform. Therefore, generated $\boldsymbol{c}_1$ and $\boldsymbol{k}^*$ has the same joint distribution as in Game 2-$(k-1)$ (i.e., semi-functional ciphertext and normal key).

When $\beta = 1$, secret key $\boldsymbol{k}^*$ generated in case (b) of step 4 or 6 is

$$\boldsymbol{k}^* = \omega \textstyle\sum_{i=1}^n v_i \boldsymbol{b}_i^* + \tau \sum_{i=1}^n (\overrightarrow{v} \cdot \overrightarrow{z}_i)\boldsymbol{b}_{n+i}^* + \boldsymbol{b}_{2n+1}^* + (\overrightarrow{v} \cdot \overrightarrow{\gamma})\boldsymbol{b}_{2n+2}^*,$$

where $\overrightarrow{\gamma} := (\gamma_1, \ldots, \gamma_n)$, $\overrightarrow{z}_i := (z_{1,i}, \ldots, z_{n,i})$. Then, ω and $\overrightarrow{v} \cdot \overrightarrow{\gamma} \in \mathbb{F}_q$ are distributed uniformly and independently from the other coefficients in $\boldsymbol{c}_1$ and $\boldsymbol{k}^*$. Since $Z = (U^{-1})^{\mathrm{T}}$ where $Z := (z_{i,j})$ and $U := (u_{i,j})$, we should verify the independence of coefficient vectors $\overrightarrow{r} := (\rho\overrightarrow{x} \cdot \overrightarrow{u}_1, \ldots, \rho\overrightarrow{x} \cdot \overrightarrow{u}_n) \in \mathbb{F}_q^n \setminus \{\overrightarrow{0}\}$ in $\boldsymbol{c}_1$ and $\overrightarrow{w} := (\tau\overrightarrow{v} \cdot \overrightarrow{z}_1, \ldots, \tau\overrightarrow{v} \cdot \overrightarrow{z}_n) \in \mathbb{F}_q^n \setminus \{\overrightarrow{0}\}$ in $\boldsymbol{k}^*$. Since $\overrightarrow{x} \cdot \overrightarrow{v} \neq 0$ from the condition given in steps 3 and 5 in Definition 17, coefficients vectors $\overrightarrow{r}$ and $\overrightarrow{w}$ are (pairwise-)independently and uniformly distributed under the condition that $\overrightarrow{r} \cdot \overrightarrow{w} \neq 0$ from Lemma 23. Since $\overrightarrow{r}, \overrightarrow{w} \xleftarrow{\mathsf{U}} \mathbb{F}_q^n$ in Game 2-k, the event that $\overrightarrow{r} \cdot \overrightarrow{w} = 0$ occurs in the game with probability $\frac{1}{q}$. Therefore, generated $\boldsymbol{c}_1$ and $\boldsymbol{k}^*$ has the same joint distribution as in Game 2-k (i.e., semi-functional ciphertext and semi-functional key), except with probability $\frac{1}{q}$. □

From Claim 2, when $\beta = 0$, the advantage of $\mathcal{A}$ in the above game is equal to that in Game 2-$(k-1)$, i.e., $\mathsf{Adv}_{\mathcal{A}}^{(2\text{-}(k-1))}(\lambda)$, and is also equal to $\mathsf{Pr}_0 := \mathsf{Pr}\left[\mathcal{B}_k(1^\lambda, \varrho) \to 1 \;\middle|\; \varrho \xleftarrow{\mathsf{R}} \mathcal{G}_0^{\mathsf{P2}}(1^\lambda, n)\right]$. When $\beta = 1$, except in the event that occurs with probability $\frac{1}{q}$, the above game is the same as Game 2-k. Hence, when $\beta = 1$, since the advantage of $\mathcal{A}$ in the above game is equal to $\mathsf{Pr}_1 := \mathsf{Pr}\left[\mathcal{B}_k(1^\lambda, \varrho) \to 1 \;\middle|\; \varrho \xleftarrow{\mathsf{R}} \mathcal{G}_1^{\mathsf{P2}}(1^\lambda, n)\right]$, $\mathsf{Adv}_{\mathcal{A}}^{(2\text{-}k)}(\lambda) \leq \mathsf{Pr}_1 + \frac{1}{q}$ from Shoup's difference lemma. Therefore, $|\mathsf{Adv}_{\mathcal{A}}^{(2\text{-}(k-1))}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(2\text{-}k)}(\lambda)| \leq |\mathsf{Pr}_0 - \mathsf{Pr}_1| + \frac{1}{q} = \mathsf{Adv}_{\mathcal{B}_k}^{\mathsf{P2}}(\lambda) + \frac{1}{q}$. This completes the proof of Lemma 25. □

Lemma 26 *For any adversary $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{(2\text{-}\nu)}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{(3)}(\lambda)$.*

Proof. To prove Lemma 26, we will show distribution $(\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \{\boldsymbol{k}^{(j)*}\}_{j=1,\ldots,\nu}, \boldsymbol{c}_1, c_2)$ in Game 2-ν and that in Game 3 are equivalent. For that purpose, we define new bases $\mathbb{D}$ of $\mathbb{V}$ and $\mathbb{D}^*$ of $\mathbb{V}^*$ as follows:

We generate random $F := (\xi_{i,s}) \xleftarrow{\mathsf{U}} \mathbb{F}_q^{n\times n}$, $\theta_i \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and set

$$\begin{aligned}
&\boldsymbol{d}_{n+i} := \boldsymbol{b}_{n+i} - \textstyle\sum_{s=1}^n \xi_{i,s}\boldsymbol{b}_s - \theta_i \boldsymbol{b}_{2n+1}, \quad \boldsymbol{d}^*_i := \boldsymbol{b}^*_i + \textstyle\sum_{s=1}^n \xi_{s,i}\boldsymbol{b}^*_{n+s} \quad \text{for } i = 1,\ldots,n,\\
&\boldsymbol{d}^*_{2n+1} := \boldsymbol{b}^*_{2n+1} + \textstyle\sum_{s=1}^n \theta_s \boldsymbol{b}^*_{n+s}.
\end{aligned}$$

Let

$$\begin{aligned}
&\overrightarrow{\boldsymbol{b}}_1 := (\boldsymbol{b}_1,\ldots,\boldsymbol{b}_n)^{\mathrm{T}},\ \overrightarrow{\boldsymbol{b}}_2 := (\boldsymbol{b}_{n+1},\ldots,\boldsymbol{b}_{2n})^{\mathrm{T}},\ \overrightarrow{\boldsymbol{b}}^*_1 := (\boldsymbol{b}^*_1,\ldots,\boldsymbol{b}^*_n)^{\mathrm{T}},\ \overrightarrow{\boldsymbol{b}}^*_2 := (\boldsymbol{b}^*_{n+1},\ldots,\boldsymbol{b}^*_{2n})^{\mathrm{T}},\\
&\overrightarrow{\boldsymbol{d}}_2 := (\boldsymbol{d}_{n+1},\ldots,\boldsymbol{d}_{2n})^{\mathrm{T}},\ \overrightarrow{\boldsymbol{d}}^*_1 := (\boldsymbol{d}^*_1,\ldots,\boldsymbol{d}^*_n)^{\mathrm{T}},\ \overrightarrow{\theta} := (\theta_1,\ldots,\theta_n)^{\mathrm{T}}.
\end{aligned}$$

That is,

$$\begin{pmatrix} \overrightarrow{\boldsymbol{b}}_1 \\ \overrightarrow{\boldsymbol{d}}_2 \\ \boldsymbol{b}_{2n+1} \end{pmatrix} := \begin{pmatrix} I_n & 0_n & 0 \\ -F & I_n & -\overrightarrow{\theta} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \overrightarrow{\boldsymbol{b}}_1 \\ \overrightarrow{\boldsymbol{b}}_2 \\ \boldsymbol{b}_{2n+1} \end{pmatrix},$$
$$\begin{pmatrix} \overrightarrow{\boldsymbol{d}}^*_1 \\ \overrightarrow{\boldsymbol{b}}^*_2 \\ \boldsymbol{d}^*_{2n+1} \end{pmatrix} := \begin{pmatrix} I_n & F^{\mathrm{T}} & 0 \\ 0_n & I_n & 0 \\ 0 & (\overrightarrow{\theta})^{\mathrm{T}} & 1 \end{pmatrix} \begin{pmatrix} \overrightarrow{\boldsymbol{b}}^*_1 \\ \overrightarrow{\boldsymbol{b}}^*_2 \\ \boldsymbol{b}^*_{2n+1} \end{pmatrix}.$$

We set

$$\begin{aligned}
\mathbb{D} &:= (\boldsymbol{b}_1,\ldots,\boldsymbol{b}_n,\boldsymbol{d}_{n+1},\ldots,\boldsymbol{d}_{2n},\boldsymbol{b}_{2n+1},\boldsymbol{b}_{2n+2},\boldsymbol{b}_{2n+3}),\\
\mathbb{D}^* &:= (\boldsymbol{d}^*_1,\ldots,\boldsymbol{d}^*_n,\boldsymbol{b}^*_{n+1},\ldots,\boldsymbol{b}^*_{2n},\boldsymbol{d}^*_{2n+1},\boldsymbol{b}^*_{2n+2},\boldsymbol{b}^*_{2n+3}).
\end{aligned}$$

We then easily verify that $\mathbb{D}$ and $\mathbb{D}^*$ are dual orthonormal, and are distributed the same as the original bases, $\mathbb{B}$ and $\mathbb{B}^*$.

Keys and challenge ciphertext $(\{\boldsymbol{k}^{(j)*}\}_{j=1,\ldots,\nu}, \boldsymbol{c}_1, c_2)$ in Game 2-ν are expressed over bases $\mathbb{B}$ and $\mathbb{B}^*$ as

$$\begin{aligned}
&\boldsymbol{k}^{(j)*} = \sigma(\textstyle\sum_{i=1}^n v_i^{(j)}\boldsymbol{b}^*_i) + \sum_{i=1}^n r_i^{(j)}\boldsymbol{b}^*_{n+i} + \boldsymbol{b}^*_{2n+1} + \eta^{(j)}\boldsymbol{b}^*_{2n+2},\\
&\boldsymbol{c}_1 = \delta_1(\textstyle\sum_{i=1}^n x_i\boldsymbol{b}_i) + \sum_{i=1}^n w_i\boldsymbol{b}_{n+i} + \zeta\boldsymbol{b}_{2n+1} + \delta_2\boldsymbol{b}_{2n+3}, \quad c_2 := g_T^{\zeta} m^{(b)}.
\end{aligned}$$

Then,

$$\begin{aligned}
\boldsymbol{k}^{(j)*} &= \sigma^{(j)}(\textstyle\sum_{i=1}^n v_i^{(j)}\boldsymbol{b}^*_i) + \sum_{s=1}^n r_s^{(j)}\boldsymbol{b}^*_{n+s} + \boldsymbol{b}^*_{2n+1} + \eta^{(j)}\boldsymbol{b}^*_{2n+2}\\
&= \sigma^{(j)}(\textstyle\sum_{i=1}^n v_i^{(j)}(\boldsymbol{d}^*_i - \sum_{s=1}^n \xi_{s,i}\boldsymbol{b}^*_{n+s})) + \sum_{s=1}^n r_s^{(j)}\boldsymbol{b}^*_{n+s} + (\boldsymbol{d}^*_{2n+1} - \textstyle\sum_{s=1}^n \theta_s\boldsymbol{b}^*_{n+s}) + \eta^{(j)}\boldsymbol{b}^*_{2n+2}\\
&= \sigma^{(j)}(\textstyle\sum_{i=1}^n v_i^{(j)}\boldsymbol{d}^*_i) + \sum_{s=1}^n (r_s^{(j)} - \sigma^{(j)}\sum_{i=1}^n v_i^{(j)}\xi_{s,i} - \theta_s)\boldsymbol{b}^*_{n+s} + \boldsymbol{d}^*_{2n+1} + \eta^{(j)}\boldsymbol{b}^*_{2n+2}\\
&= \sigma^{(j)}(\textstyle\sum_{i=1}^n v_i^{(j)}\boldsymbol{d}^*_i) + \sum_{s=1}^n \gamma_s^{(j)}\boldsymbol{b}^*_{n+s} + \boldsymbol{d}^*_{2n+1} + \eta^{(j)}\boldsymbol{b}^*_{2n+2},
\end{aligned}$$

where

$$\gamma_s^{(j)} := r_s^{(j)} - \sigma^{(j)}\textstyle\sum_{i=1}^n v_i^{(j)}\xi_{s,i} - \theta_s,$$

which are uniformly, independently distributed since $r_s^{(j)} \xleftarrow{\mathsf{U}} \mathbb{F}_q$.

$$\begin{aligned}
\boldsymbol{c}_1 &= \delta_1(\textstyle\sum_{i=1}^n x_i\boldsymbol{b}_i) + \sum_{t=1}^n w_t\boldsymbol{b}_{n+t} + \zeta\boldsymbol{b}_{2n+1} + \delta_2\boldsymbol{b}_{2n+3}\\
&= \delta_1(\textstyle\sum_{i=1}^n x_i\boldsymbol{b}_i) + \sum_{t=1}^n w_t(\boldsymbol{d}_{n+t} + \sum_{s=1}^n \xi_{t,s}\boldsymbol{b}_s + \theta_t\boldsymbol{b}_{2n+1}) + \zeta\boldsymbol{b}_{2n+1} + \delta_2\boldsymbol{b}_{2n+3}\\
&= \textstyle\sum_{i=1}^n (\delta_1 x_i + \sum_{t=1}^n w_t\xi_{t,i})\,\boldsymbol{b}_i + \sum_{t=1}^n w_t\boldsymbol{d}_{n+t} + (\zeta + \sum_{t=1}^n w_t\theta_t)\,\boldsymbol{b}_{2n+1} + \delta_2\boldsymbol{b}_{2n+3}\\
&= \textstyle\sum_{i=1}^n x'_i\boldsymbol{b}_i + \sum_{t=1}^n w_t\boldsymbol{d}_{n+t} + \zeta'\boldsymbol{b}_{2n+1} + \delta_2\boldsymbol{b}_{2n+3},
\end{aligned}$$

where

$$x'_i := \delta_1 x_i + \sum_{t=1}^{n} w_t \xi_{t,i}, \quad \zeta' := \zeta + \sum_{t=1}^{n} w_t \theta_t,$$

which are uniformly, independently distributed since $(w_1, \ldots, w_n) \xleftarrow{\mathsf{U}} \mathbb{F}_q^n \setminus \{\overrightarrow{0}\}, (\xi_{t,i}) \xleftarrow{\mathsf{U}} \mathbb{F}_q^{n \times n}$, $\theta_i \xleftarrow{\mathsf{U}} \mathbb{F}_q$.

In the light of the adversary's view, both $(\mathbb{B}, \mathbb{B}^*)$ and $(\mathbb{D}, \mathbb{D}^*)$ are consistent with public key $\mathsf{pk} := (1^\lambda, \mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}})$. Therefore, $\{\boldsymbol{k}^{(j)*}\}_{j=1,\ldots,\nu}$ and $\boldsymbol{c}_1$ above can be expressed as keys and ciphertext in two ways, in Game 2-ν over bases $(\mathbb{B}, \mathbb{B}^*)$ and in Game 3 over bases $(\mathbb{D}, \mathbb{D}^*)$. Thus, Game 2-$\nu$ can be conceptually changed to Game 3. □

Lemma 27 *For any adversary $\mathcal{A}$, $\mathsf{Adv}^{(3)}_{\mathcal{A}}(\lambda) = 0$.*

Proof. The value of b is independent from the adversary's view in Game 3. Hence, $\mathsf{Adv}^{(3)}_{\mathcal{A}}(\lambda) = 0$. □

3.6 The Proposed HPE Scheme

The definition of HPE and key idea for the proposed HPE (and the correctness of the HPE) are given in Appendices B.5 and B.6.

3.6.1 Construction

$\mathsf{Setup}(1^\lambda, \overrightarrow{\mu} := (n, d; \mu_1, \ldots, \mu_d)) :$ $(\mathsf{param}_{\mathbb{V}}, \mathbb{B}, \mathbb{B}^*) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^\lambda, 2n+3)$,
$\widehat{\mathbb{B}} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n, \boldsymbol{b}_{2n+1}, \boldsymbol{b}_{2n+3})$, $\mathsf{sk} := \mathbb{B}^*$, $\mathsf{pk} := (1^\lambda, \mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}})$,
return sk, pk.

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{sk}, (\overrightarrow{v}_1, \ldots, \overrightarrow{v}_\ell) := ((v_1, \ldots, v_{\mu_1}), \ldots, (v_{\mu_{\ell-1}+1}, \ldots, v_{\mu_\ell})) :$

$\sigma_{\mathsf{dec},t}, \eta_{\mathsf{dec}}, \ \sigma_{\mathsf{ran},j,t}, \eta_{\mathsf{ran},j}$ $(j = 1, .., \ell+1)$, $\sigma_{\mathsf{del},j,t}, \eta_{\mathsf{del},j}$ $(j = 1, .., n)$, $\psi \xleftarrow{\mathsf{U}} \mathbb{F}_q$
for $t = 1, \ldots, \ell$,
$\boldsymbol{k}^*_{\ell,\mathsf{dec}} := \sum_{t=1}^{\ell} \sigma_{\mathsf{dec},t} (\sum_{i=\mu_{t-1}+1}^{\mu_t} v_i \boldsymbol{b}^*_i) + \boldsymbol{b}^*_{2n+1} + \eta_{\mathsf{dec}} \boldsymbol{b}^*_{2n+2}$,
$\boldsymbol{k}^*_{\ell,\mathsf{ran},j} := \sum_{t=1}^{\ell} \sigma_{\mathsf{ran},j,t} (\sum_{i=\mu_{t-1}+1}^{\mu_t} v_i \boldsymbol{b}^*_i) + \eta_{\mathsf{ran},j} \boldsymbol{b}^*_{2n+2}$ for $j = 1, \ldots, \ell+1$,
$\boldsymbol{k}^*_{\ell,\mathsf{del},j} := \sum_{t=1}^{\ell} \sigma_{\mathsf{del},j,t} (\sum_{i=\mu_{t-1}+1}^{\mu_t} v_i \boldsymbol{b}^*_i) + \psi \boldsymbol{b}^*_j + \eta_{\mathsf{del},j} \boldsymbol{b}^*_{2n+2}$
for $j = \mu_\ell + 1, \ldots, n$,
return $\overrightarrow{\boldsymbol{k}}^*_\ell := (\boldsymbol{k}^*_{\ell,\mathsf{dec}}, \boldsymbol{k}^*_{\ell,\mathsf{ran},1}, \ldots, \boldsymbol{k}^*_{\ell,\mathsf{ran},\ell+1}, \boldsymbol{k}^*_{\ell,\mathsf{del},\mu_\ell+1}, \ldots, \boldsymbol{k}^*_{\ell,\mathsf{del},n})$.

$\mathsf{Enc}(\mathsf{pk}, m \in \mathbb{G}_T, (\overrightarrow{x}_1, \ldots, \overrightarrow{x}_\ell) := ((x_1, \ldots, x_{\mu_1}), \ldots, (x_{\mu_{\ell-1}+1}, \ldots, x_{\mu_\ell})) :$

$(\overrightarrow{x}_{\ell+1}, \ldots, \overrightarrow{x}_d) \xleftarrow{\mathsf{U}} \mathbb{F}_q^{\mu_{\ell+1}-\mu_\ell} \times \cdots \times \mathbb{F}_q^{n-\mu_{d-1}}, \quad \delta_1, \ldots, \delta_\ell, \delta_{2n+3}, \zeta \xleftarrow{\mathsf{U}} \mathbb{F}_q,$

$\boldsymbol{c}_1 := \sum_{t=1}^{\ell} \delta_t (\sum_{i=\mu_{t-1}+1}^{\mu_t} x_i \boldsymbol{b}_i) + \zeta \boldsymbol{b}_{2n+1} + \delta_{2n+3} \boldsymbol{b}_{2n+3}, \quad c_2 := g_T^{\zeta} m,$

return $(\boldsymbol{c}_1, c_2)$.

$\mathsf{Dec}(\mathsf{pk}, \boldsymbol{k}^*_{\ell,\mathsf{dec}}, \boldsymbol{c}_1, c_2) : \ m' := c_2 / e(\boldsymbol{c}_1, \boldsymbol{k}^*_{\ell,\mathsf{dec}}),$

return m'.

$\mathsf{Delegate}_\ell(\mathsf{pk}, \overrightarrow{\boldsymbol{k}}^*_\ell, \overrightarrow{v}_{\ell+1} := (v_{\mu_\ell+1}, \ldots, v_{\mu_{\ell+1}})) :$

$\alpha_{\mathsf{dec},t}, \sigma_{\mathsf{dec}}, \ \alpha_{\mathsf{ran},j,t}, \sigma_{\mathsf{ran},j} \ (j = 1, .., \ell+2), \ \alpha_{\mathsf{del},j,t}, \sigma_{\mathsf{del},j} \ (j = 1, .., n), \ \psi' \xleftarrow{\mathsf{U}} \mathbb{F}_q$
for $t = 1, \ldots, \ell+1$,

$\boldsymbol{k}^*_{\ell+1,\mathsf{dec}} := \boldsymbol{k}^*_{\ell,\mathsf{dec}} + \sum_{t=1}^{\ell+1} \alpha_{\mathsf{dec},t} \boldsymbol{k}^*_{\ell,\mathsf{ran},t} + \sigma_{\mathsf{dec}} (\sum_{i=\mu_\ell+1}^{\mu_{\ell+1}} v_i \boldsymbol{k}^*_{\ell,\mathsf{del},i}),$

$\boldsymbol{k}^*_{\ell+1,\mathsf{ran},j} := \sum_{t=1}^{\ell+1} \alpha_{\mathsf{ran},j,t} \boldsymbol{k}^*_{\ell,\mathsf{ran},t} + \sigma_{\mathsf{ran},j} (\sum_{i=\mu_\ell+1}^{\mu_{\ell+1}} v_i \boldsymbol{k}^*_{\ell,\mathsf{del},i})$ for $j = 1, .., \ell+2,$

$\boldsymbol{k}^*_{\ell+1,\mathsf{del},j} := \sum_{t=1}^{\ell+1} \alpha_{\mathsf{del},j,t} \boldsymbol{k}^*_{\ell,\mathsf{ran},t} + \sigma_{\mathsf{del},j} (\sum_{i=\mu_\ell+1}^{\mu_{\ell+1}} v_i \boldsymbol{k}^*_{\ell,\mathsf{del},i}) + \psi' \boldsymbol{k}^*_{\ell,\mathsf{del},j}$
for $j = \mu_{\ell+1} + 1, \ldots, n$,

return $\overrightarrow{\boldsymbol{k}}^*_{\ell+1} := (\boldsymbol{k}^*_{\ell+1,\mathsf{dec}}, \boldsymbol{k}^*_{\ell+1,\mathsf{ran},1}, .., \boldsymbol{k}^*_{\ell+1,\mathsf{ran},\ell+2}, \boldsymbol{k}^*_{\ell+1,\mathsf{del},\mu_{\ell+1}+1}, .., \boldsymbol{k}^*_{\ell+1,\mathsf{del},n})$.

Remark: A PE scheme with general delegation is given in Appendix B.8.

3.6.2 Security

Theorem 28 *The proposed HPE scheme is adaptively attribute-hiding against chosen plaintext attacks under the n-eDDH assumption. For any adversary $\mathcal{A}$, there exist probabilistic machines, $\mathcal{C}_0$ and $\mathcal{C}_{(k,j)}$ ($k = 1, \ldots, \nu$; $j = 1, \ldots, n+1$) whose running times are essentially the same as that of $\mathcal{A}$, such that for any security parameter λ,*

$$\mathsf{Adv}^{\mathsf{HPE,AH}}_{\mathcal{A}}(\lambda) < \mathsf{Adv}^{n\text{-}\mathsf{eDDH}}_{\mathcal{C}_0}(\lambda) + \sum_{k=1}^{\nu} \sum_{j=1}^{n+1} \mathsf{Adv}^{n\text{-}\mathsf{eDDH}}_{\mathcal{C}_{(k,j)}}(\lambda) + \frac{(n+4)\nu}{q},$$

where ν is the maximum number of adversary $\mathcal{A}$'s key queries.

The proof is given in Appendix B.7.

References

[1] S. Al-Riyami, J. Malone-Lee, and N. Smart. Escrow-free encryption supporting cryptographic workflow. In *Int. J. Inf. Sec.*, volume 5, pages 217–229, 2006.

[2] W. Bagga, R. Molva, and S. Crosta. Policy-based encryption schemes from bilinear pairings. In *ASIACCS*, page 368, 2006.

[3] M. Barbosa and P. Farshim. Secure cryptographic workflow in the standarad model. In *INDOCRYPT*, pages 379–393, 2006.

[4] A. Beimel. Secure schemes for secret sharing and key distribution. PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.

[5] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2007.

[6] D. Boneh and X. Boyen. Efficient selective-id secure identity based encryption without random oracles. In *EUROCRYPT*, pages 223 – 238, 2004.

[7] D. Boneh and X. Boyen. Secure identity based encryption without random oracles. In *CRYPTO*, pages 443–459, 2004.

[8] D. Boneh, X. Boyen, and E. Goh. Hierarchical identity based encryption with constant size ciphertext. In *EUROCRYPT*, pages 440–456, 2005.

[9] D. Boneh and M. Franklin. Identity based encryption from the weil pairing. In *CRYPTO*, pages 213–229, 2001.

[10] D. Boneh, E. Goh, and K. Nissim. Evaluating 2-dnf formulas on ciphertexts. In *TCC*, pages 325–342, 2005.

[11] D. Boneh and M. Hamburg. Generalized identity based and broadcast encryption scheme. In *ASIACRYPT*, pages 455–470, 2008.

[12] D. Boneh and J. Katz. Improved efficiency for cca-secure cryptosystems built using identity based encryption. In *RSA-CT*, 2005.

[13] D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In *TCC*, pages 535–554, 2007.

[14] X. Boyen and B. Waters. Anonymous hierarchical identity-based encryption (without random oracles). In *CRYPTO*, pages 290–307, 2006.

[15] R. Bradshaw, J. Holt, and K. Seamons. Concealing complex policies with hidden credentials. In *CCS*, pages 146–157, 2004.

[16] R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. In *EUROCRYPT*, pages 255–271, 2003.

[17] R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT*, 2004.

[18] M. Chase. Multi-authority attribute based encryption. In *TCC*, pages 515–534, 2007.

[19] L. Cheung and C. Newport. Provably secure ciphertext policy abe. In *CCS*, pages 456–465, 2007.

[20] C. Cocks. An identity based encryption scheme based on quadratic residues. In *Proceedings of the 8th IMA International Conference on Cryptography and Coding*, pages 26–28, 2001.

[21] David Mandell Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In *EUROCRYPT*, 2010.

[22] C. Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT*, pages 445–464, 2006.

[23] C. Gentry and S. Halevi. Hierarchical identity based encryption with polynomially many levels. In *TCC*, pages 437–456, 2009.

[24] C. Gentry and A. Silverberg. Hierarchical id-based cryptography. In *ASIACRYPT*, pages 548–566, 2002.

[25] V. Goyal, A. Jain, O. Pandey, and A. Sahai. Bounded ciphertext policy attribute-based encryption. In *ICALP*, 2008.

[26] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute Based Encryption for Fine-Grained Access Conrol of Encrypted Data. In *CCS*, 2006.

[27] J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *EUROCRYPT*, pages 415–432, 2008.

[28] J. Horwitz and B. Lynn. Toward hierarchical identity-based encryption. In *EUROCRYPT*, pages 466–481, 2002.

[29] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT*, pages 146–162, 2008.

[30] A. Lewko and B. Waters. New techniques for dual system encryption and fully secure hibe with short ciphertexts. In *TCC*, 2010.

[31] G. Miklau and D. Suciu. Controlling access to published data using cryptography. In *VLDB*, pages 898–909, 2003.

[32] T. Okamoto and K. Takashima. Homomorphic encryption and signatures from vector decomposition. In *Pairing*, pages 57–74, 2008.

[33] T. Okamoto and K. Takashima. Hierarchical predicate encryption for inner-products. In *ASIACRYPT*, 2009.

[34] Rafail Ostrovksy, Amit Sahai, and Brent Waters. Attribute Based Encryption with Non-Monotonic Access Structures. In *CCS*, 2007.

[35] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters. Secure attribute-based systems. In *CCS*, pages 99–112, 2006.

[36] A. Sahai and B. Waters. Fuzzy Identity Based Encryption. In *EUROCRYPT*, pages 457–473, 2005.

[37] A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.

[38] E. Shi and B. Waters. Delegating capabilities in predicate encryption systems. In *Automata, Languages and Programming*, pages 560–578, 2008.

[39] N. Smart. Access control using pairing based cryptography. In *CT-RSA*, pages 111–121, 2003.

[40] B. Waters. Efficient identity-based ecnryption without random oracles. In *EUROCRYPT*, pages 114–127, 2005.

[41] B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. Cryptology ePrint Archive, Report 2008/290, 2008.

[42] B. Waters. Dual system encryption: realizing fully secure ibe and hibe under simple assumptions. In *CRYPTO*, pages 619–636, 2009.

A Appendices for Attribute-Based Encryption

A.1 KP-ABE

The same techniques we applied above to obtain a fully secure CP-ABE system also yield a fully secure KP-ABE system. Our system is very closely related to the selectively secure KP-ABE systems in [26].

A.1.1 Definition

A KP-ABE system is very similar to a CP-ABE system, except that keys are now associated with access structures and ciphertexts are associated with sets of attributes. A KP-ABE system consists of four algorithms:

Setup$(\lambda, U) \rightarrow PK, MSK$ The setup algorithm takes in the security parameter λ and the attribute universe description U. It outputs the public parameters PK and a master secret key MSK.

Encrypt$(M, PK, S) \rightarrow CT$ The encryption algorithm takes in a message M, the public parameters, and a set of attributes S. It outputs a ciphertext CT.

KeyGen$(\mathbb{A}, MSK, PK) \rightarrow SK$ The key generation algorithm takes in an access structure $\mathbb{A}$, the master secret key MSK, and the public parameters PK. It outputs a secret key SK.

Decrypt($CT, SK, PK) \rightarrow M$ The decryption algorithm takes in a ciphertext encrypted under a set of attributes S, a secret key for an access structure $\mathbb{A}$, and the public parameters. It will output the message M if S satisfies $\mathbb{A}$.

The security definition for KP-ABE is essentially the same as for CP-ABE: first the challenger gives the attacker the public parameters. Then the attacker can make private key queries (phase 1). In the challenge phase, the attacker provides two messages M_0, M_1 and a set of attributes S^* that does not satisfy any of the private key queries made in phase 1. The challenger flips a random coin $\beta \in \{0, 1\}$ and gives the attacker an encryption of M_β under S^*. In phase 2, the attacker can again make private key queries, subject to the restriction that none can be satisfied by S^*. Finally, the attacker outputs a guess β' for β. The advantage of the attacker is $Pr[\beta = \beta'] - \frac{1}{2}$. Selective security is defined by having the attacker declare S^* before seeing the public parameters.

A.1.2 Construction

We now construct our one-use KP-ABE system in composite order groups of order $N = p_1 p_2 p_3$. It is closely related to the selectively secure KP-ABE systems in [26], and our proof of security will be similar to the proof for our CP-ABE scheme. We again rely on Assumptions 1, 2, and 3 and restrict to case where each attribute can be used at most once in labeling the rows of an LSSS matrix. The matrices are now associated with keys and sets of attributes are associated with ciphertexts. The same general transformation can be used to yield a secure scheme where attributes can be used more than once.

Our KP-ABE system consists of four algorithms:

Setup$(\lambda) \to PK, MSK$ The setup algorithm chooses a bilinear group G of order $N = p_1p_2p_3$ (3 distinct primes). We let G_{p_i} denote the subgroup of order p_i in G. It chooses a random $\alpha \in \mathbb{Z}_N$ and $g \in G_{p_1}$. For each attribute i, it chooses a random $s_i \in \mathbb{Z}_N$. The public parameters are $N, g, e(g,g)^\alpha, T_i = g^{s_i} \forall i$. The master secret key is α and a generator X_3 of G_{p_3}.

Encrypt$(M, S, PK) \to CT$ The encryption algorithm chooses a random $s \in \mathbb{Z}_N$. It sets the ciphertext to be:

$$C = Me(g,g)^{\alpha s},\ C_0 = g^s,\ C_i = T_i^s\ \forall i \in S.$$

(This implicitly includes S.)

KeyGen$((A, \rho), MSK, PK) \to SK$ A is a matrix and ρ is map from each row x of A to an attribute $\rho(x)$. The key generation algorithm chooses a random vector u such that $1 \cdot u = \alpha$. (Here, 1 denotes the vector with the first entry equal to 1 and the rest equal to 0). For each row A_x of A, it chooses a random $r_x \in \mathbb{Z}_N$, random elements $W_x, V_x \in G_{p_3}$, and sets the secret key to be:

$$K_x^1 = g^{A_x \cdot u} T_{\rho(x)}^{r_x} W_x,$$

$$K_x^2 = g^{r_x} V_x.$$

Decrypt$(CT, PK, SK) \to M$ We let S denote the set of attributes associated to CT and let A, ρ denote the matrix and row labeling associated with SK. If S satisfies A, the decryption algorithm computes constants ω_x such that $\sum_{\rho(x) \in S} \alpha_x A_x = 1$. It then computes:

$$\prod_{\rho(x) \in S} \frac{e(C_0, K_x^1)^{\omega_x}}{e(C_{\rho(x)}, K_x^2)^{\omega_x}} = \prod_{\rho(x) \in S} \frac{e(g,g)^{s\omega_x A_x \cdot u} e(g, T_{\rho(x)})^{s r_x \omega_x}}{e(g, T_{\rho(x)})^{s r_x \omega_x}}$$

$$= e(g,g)^{s \sum_{\rho(x) \in S} \omega_x A_x \cdot u} = e(g,g)^{s\alpha}.$$

The message can then be recovered as $C / e(g,g)^{s\alpha}$.

A.1.3 Security

To prove that our system is fully secure, we must first define semi-functional keys and ciphertexts.

Semi-functional Ciphertext A semi-functional ciphertext is formed as follows. We let g_2 denote a generator of G_{p_2} and c a random exponent. We also choose random values z_i. Then:

$$C_0 = g^s g_2^c,\ C_i = T_i^s g_2^{c z_i}\ \forall i \in S.$$

Semi-functional Key A semi-functional key will take on one of two forms. A semi-functional key of type 1 is formed as follows. A random vector u_2 is chosen (so $u_2 \cdot 1$ is random), and we set $\delta_x = A_x \cdot u_2$. Additionally, random exponents γ_x are chosen. The key is then defined as:

$$K_x^1 = g^{A_x \cdot u} T_{\rho(x)}^{r_x} W_x g_2^{\delta_x + \gamma_x z_{\rho(x)}},$$

$$K_x^2 = g^{r_x} V_x g_2^{\gamma_x}.$$

A semi-functional key of type 2 is formed similarly, except without the terms $g_2^{\gamma_x z_{\rho(x)}}$ and $g_2^{\gamma_x}$ (one could also interpret this as setting $\gamma_x = 0$):

$$K_x^1 = g^{A_x \cdot u} T_{\rho(x)}^{r_x} W_x g_2^{\delta_x},$$

$$K_x^2 = g^{r_x} V_x.$$

We note that when we use a semi-functional key to decrypt a semi-functional ciphertext, we are left with an additional term:

$$e(g_2, g_2)^{\sum_{\rho(x) \in S} c\omega_x \delta_x} = e(g_2, g_2)^{cu_2 \cdot 1}.$$

We also note that these values z_i are common to semi-functional ciphertexts and semi-functional keys of type 1. We call a semi-functional key a *nominally* semi-functional key if $u_2 \cdot 1 = 0$. Notice that if a nominally semi-functional key is used to decrypt a semi-functional ciphertext, decryption will still work.

We will prove adaptive security of our system from our assumptions using a sequence of games. The first game, Game_{Real}, is the real security game (the challenge ciphertext and all are normal). In Game_0, all keys are normal, but the challenge ciphertext is semi-functional. In $\text{Game}_{k,1}$, the first $k-1$ keys are semi-functional of type 2, key k is semi-functional of type 1, and keys $> k$ are normal. In $\text{Game}_{k,2}$, the first k keys are semi-functional of type 2 and keys $> k$ are normal. Notice that in $\text{Game}_{q,2}$ (where q is the number of key queries), all keys are semi-functional of type 2. In Game_{Final}, the keys are all semi-functional of type 2, and the challenge ciphertext is a semi-functional encryption of a random message. We will prove this games are indistinguishable in the following lemmas.

Lemma 29 *Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $\text{Game}_{Real}Adv_{\mathcal{A}} - \text{Game}_0 Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 1.*

Proof. $\mathcal{B}$ is given $\{g, X_3, T\}$. It will simulate either Game_{Real} or Game_0 with $\mathcal{A}$. The public parameters are formed by choosing exponents α, s_i randomly. $\mathcal{B}$ gives these to $\mathcal{A}$. $\mathcal{B}$ can respond to key requests by running the usual key generation algorithm to make normal keys because it knows the MSK.

To form the challenge ciphertext for a set of attributes S, $\mathcal{B}$ implicitly sets s so that g^s is the part of T in the G_{p_1} subgroup (i.e. T is the product of g^s and possibly an element of G_{p_2}:

$$C = M_\beta e(g,g)^{s\alpha} = Me(g,T)^\alpha,\ C_0 = T,\ C_i = T^{s_i}\ \forall i \in S.$$

We note that this sets $z_i = s_i$. However, the values of s_i modulo p_1 are uncorrelated from the values of z_i modulo p_2 by the Chinese Remainder Theorem. If $T = g^s$, this is a properly distributed normal ciphertext. If $T = g^s X_2$ (for $X_2 \in G_{p_2}$), this is a properly distributed semi-functional ciphertext. Thus, $\mathcal{B}$ can use the output of $\mathcal{A}$ to gain advantage ϵ in breaking Assumption 1. □

Lemma 30 *Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $\text{Game}_{k-1,2}Adv_{\mathcal{A}} - \text{Game}_{k,1}Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage negligibly close to ϵ in breaking Assumption 2.*

Proof. $\mathcal{B}$ receives $\{g, X_3, g^s X_2, Y_2 Y_3, T\}$. It will simulate either $\text{Game}_{k-1,2}$ or $\text{Game}_{k,1}$ with $\mathcal{A}$. It begins by sending $\mathcal{A}$ the public parameters $N, g, X_3, e(g,g)^\alpha, g^{s_i}$, where it chooses α and the values s_i randomly.

To form the challenge ciphertext for a set of attributes S, $\mathcal{B}$ sets:

$$C = M_\beta e(g, g^s X_2)^\alpha,\ C_0 = g^s X_2,\ C_i = (g^s X_2)^{s_i}\ \forall i \in S.$$

This implicitly sets $s_i = z_i$ modulo N, but again these values are actually uncorrelated because s_i is used as a modulo p_1 value and z_i is used as a modulo p_2 value.

To form normal keys for queries $> k$, $\mathcal{B}$ can use its knowledge of the MSK and employ the regular key generation algorithm. To create semi-functional keys of type 2 for queries $< k$, $\mathcal{B}$ chooses a random vector u such that $u \cdot 1 = \alpha$, a random vector u_2', random values $r_x \in \mathbb{Z}_p$, and random group elements $W_x, V_x \in G_{p_3}$. The semi-functional key can then be defined as:

$$K_x^1 = g^{A_x \cdot u} T_{\rho(x)}^{r_x} W_x (Y_2 Y_3)^{A_x \cdot u_2'},$$

$$K_x^2 = g^{r_x} V_x.$$

We note that for $Y_2 = g_2^c$, the u_2 in our description of semi-functional keys above now corresponds to $u_2 = c u_2'$.

For the k^{th} key, $\mathcal{B}$ will make a key that is either normal or nominally semi-functional of type 1, depending on the value of T in the challenge. We will later argue that if it is a nominally semi-functional key, it will still have the distribution of a regular semi-functional key of type 1 in the view of an attacker, since it is not a key capable of decrypting the challenge ciphertext.

To form the k^{th} key for (A, ρ), $\mathcal{B}$ chooses a random vector u_2 such that $u_2 \cdot 1 = 0$ and a random vector u' such that $u' \cdot 1 = \alpha$. It implicitly sets $u = r u_2 + u'$, where g^r is the G_{p_1} part of T. Values γ_x, V_x, W_x are chosen randomly.

$\mathcal{B}$ computes:

$$K_x^1 = g^{A_x \cdot u'} (T)^{A_x \cdot u_2} T^{\gamma_x z_{\rho(x)}} W_x,\ K_x^2 = T^{\gamma_x} V_x.$$

We note that this sets $r_x = r\gamma_x$, but this is acceptable because r_x is used as a modulo p_1 value and γ_x is used as a modulo p_2 value. Depending on the value of T, this is either a normal key or a nominally semi-functional key of type 1.

Now we must argue that if the attacker has not asked for a k^{th} key that can decrypt the challenger ciphertext, then the k^{th} key will seem properly distributed in the attacker's view (the fact that 0 is being shared in the G_{p_2} subgroup should be information-theorectically hidden). We recall our restriction that attributes are used only once in the labeling ρ of the matrix rows in each key. We consider an attribute i that does not belong to S, the set of attributes for the challenge ciphertext. We note that value z_i does not appear anywhere except possibly in the k^{th} key, since any other semi-functional keys given out are of type 2.

We are assuming that the k^{th} key cannot decrypt the challenge ciphertext. This means that the vector 1 is not in the rowspace R of the corresponding matrix A when we restrict to rows x such that $x \in S$. We may assume this holds modulo p_2 (see earlier remarks). Thus, there is a vector w such that w is orthogonal to R, and not orthogonal to $(1, 0, \ldots, 0)$ (modulo p_2). We fix a basis including w, and we write u_2 as $u_2 = fw + u_2''$ modulo p_2 for $f \in \mathbb{Z}_{p_2}$ and u_2'' in the span of the other basis elements (note that u_2'' is uniformly distributed in this space). We note that u_2'' reveals no information about f. We note that $(1, 0, \ldots, 0) \cdot u_2$ cannot be determined from u_2'' alone, some information about f is needed. The shares for rows with $x \in S$ only reveal information about u_2''.

The only places fw appears are in equations of the form:

$$A_x \cdot u_2 + \gamma_x z_{\rho(x)},$$

where the $\rho(x)$'s are each *unique* attributes not appearing in the challenge ciphertext. As long as each γ_x is not congruent to 0 modulo p_2, each of these equations introduces a new unknown $z_{\rho(x)}$ that appears nowhere else, so no information about f can be learned by the attacker. More precisely, for each potential value of $u_{2,1}$, there are an equal number of solutions to these equations, so each value is equally likely. Hence, the value being shared in the G_{p_2} subgroup is information-theoretically hidden, as long as none of the γ_x are 0 modulo p_2. The probability that any of the γ_x values are 0 modulo p_2 is negligible. Thus, the ciphertext and key k are properly distributed in the attacker's view with probability negligible close to 1.

Therefore, depending on the value of T, $\mathcal{B}$ has properly simulated either $\text{Game}_{k-1,2}$ or $\text{Game}_{k,1}$ with probability negligibly close to 1. Hence it can use the output of $\mathcal{A}$ to gain advantage negligibly close to ϵ in breaking Assumption 2. □

Lemma 31 *Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $Game_{k,1}Adv_{\mathcal{A}} - Game_{k,2}Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 2.*

Proof. $\mathcal{B}$ receives $\{g, X_3, g^s X_2, Y_2 Y_3, T\}$. It will simulate either $\text{Game}_{k,1}$ or $\text{Game}_{k,2}$ with $\mathcal{A}$. It begins by sending $\mathcal{A}$ the public parameters $N, g, X_3, e(g,g)^\alpha, g^{s_i}$, where it chooses α and the values s_i randomly. It will implicitly set $z_i = s_i$.

To form the challenge ciphertext for a set of attributes S, $\mathcal{B}$ sets:

$$C = M_\beta e(g^s X_2, g)^\alpha,\ C_0 = g^s X_2,\ C_i = (g^s X_2)^{s_i}\ \forall i \in S.$$

To form normal keys for queries $> k$, $\mathcal{B}$ can use its knowledge of the MSK and employ the regular key generation algorithm. To create semi-functional keys of type 2 for queries $< k$, $\mathcal{B}$ can choose a random vector u such that $u \cdot 1 = \alpha$, a random vector u_2', random values r_x, and random group elements $W_x, V_x \in G_{p_3}$. The semi-functional key can then be defined as:

$$K_x^1 = g^{A_x \cdot u} T_{\rho(x)}^{r_x} W_x (Y_2 Y_3)^{A_x \cdot u_2'},$$

$$K_x^2 = g^{r_x} V_x.$$

We note that for $Y_2 = g_2^c$, the u_2 in our description of semi-functional keys above now corresponds to $u_2 = cu_2'$.

For the k^{th} key, $\mathcal{B}$ will either make a semi-functional key of type 1 or one of type 2, depending on the value of T in the assumption. $\mathcal{B}$ first chooses a random vector u such that $u \cdot 1 = \alpha$ and a random vector u_2. Part of the K_x^1 value can then be computed as:

$$g^{A_x \cdot u} (Y_2 Y_3)^{A_x \cdot u_2}.$$

Next, $\mathcal{B}$ chooses random values γ_x and implicitly sets $r_x = r\gamma_x$, where g^r is the G_{p_1} part of T. Also, W_x is chosen randomly in G_{p_3}. $\mathcal{B}$ can then compute the rest of K_x^1 as:

$$T^{\gamma(x) s_{\rho(x)}} W_x.$$

Putting it together, we have:

$$K_x^1 = g^{A_x \cdot u} (Y_2 Y_3)^{A_x \cdot u_2'} T^{\gamma_x s_{\rho(x)}} W_x.$$

Similarly, we choose V_x randomly in G_{p_3} and set

$$K_x^2 = T^{\gamma_x} V_x.$$

If $T \in G$, this is a properly distributed semi-functional key of type 1. If $T \in G_{p_1 p_3}$, this is a properly distributed semi-functional key of type 2. Hence $\mathcal{B}$ can use the output of $\mathcal{A}$ to gain advantage ϵ in breaking Assumption 2. $\square$

Lemma 32 *Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $Game_{q,2} Adv_{\mathcal{A}} - Game_{Final} Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 3.*

Proof. $\mathcal{B}$ receives g, $g^\alpha X_2$, X_3, $g^s Y_2$, Z_2, T. It will simulate $\text{Game}_{q,2}$ or Game_{Final} with $\mathcal{A}$. $\mathcal{B}$ begins by setting the public parameters $N, g, X_3, e(g,g)^\alpha = e(g, g^\alpha X_2)$. It chooses values $s_i = z_i$ randomly.

To form the challenge ciphertext for a set of attributes S, $\mathcal{B}$ computes:

$$C = M_\beta T,\ C_0 = g^s X_2,\ C_i = (g^s X_2)^{s_i}\ \forall i \in S.$$

If $T = e(g,g)^{\alpha s}$, this will be a semi-functional encryption of M_β. If T is random, this will be a semi-functional encryption of a random message and this will give no information about β to the attacker.

To form a semi-functional key of type 2 for (A, ρ) (where A has ℓ columns), $\mathcal{B}$ chooses $u_1, \ldots, u_{\ell-1}$ randomly and implicity sets $u_\ell = \alpha - \sum_{i=1}^{\ell-1} u_i$ (so that $1 \cdot u = \alpha$). $\mathcal{B}$ chooses a random vector u_2 of length ℓ. It chooses W_x randomly in G_{p_3}, r_x randomly in $\mathbb{Z}_N$, and V_x randomly in G_{p_3}. It sets:

$$K_x^1 = g^{\sum_{i=1}^{\ell-1} A_{x,i} u_i - A_{x,\ell} u_i} (g^\alpha X_2)^{A_{x,\ell}} g^{s_{\rho(x)} r_x} Z_2^{A_x \cdot u_2} W_x,$$

$$K_x^2 = g^{r_x} V_x.$$

Notice that for $X_2 = g_2^c, Z_2 = g_2^d$, in the G_{p_2} subgroup this will be $g_2^{\delta_x}$ for $\delta_x = A_x \cdot u_2'$ where $u_2' = d u_2$ except with a c added in the last coordinate. This yields a properly distributed semi-functional key of type 2. $\mathcal{B}$ can use the output of $\mathcal{A}$ to gain advantage ϵ in breaking Assumption 3. $\square$

We have now proven the following theorem:

Theorem 33 *If Assumptions 1, 2, and 3 hold, then our KP-ABE system is secure.*

Proof. If Assumptions 1, 2, and 3 hold, then we have shown by the previous lemmas that the real security game is indistinguishable from Game_{Final}, in which the value of β is information-theoretically hidden from the attacker. Hence the attacker cannot attain a non-negligible advantage in breaking the KP-ABE system. $\square$

A.1.4 Expanding to Multi-Use

To build a fully secure KP-ABE system where each attribute can be used up to k times in the row labeling of an access matrix, we apply the encoding technique of Section 2.2. We compare the resulting k-use system to the one-system in Table 2. We let $|U|$ denote the number of attributes before the encoding is applied, $|S|$ denote the number of attributes associated with a ciphertext, and ℓ denote the number of rows in an access matrix associated with a key. All sizes refer to the number of group elements.

	size of PK	size of SK	size of CT
one-use	$\|U\|+2$	2ℓ	$\|S\|+2$
k-use	$k\|U\|+2$	2ℓ	$k\|S\|+2$

Table 2: Comparison of one-use and k-use KP-ABE systems

A.2 Generic Security of Our Complexity Assumptions

We now prove our three complexity assumptions hold in the generic group model, as long as it is hard to find a nontrivial factor of the group order, n. We adopt the notation of [29] to express our assumptions. We fix generators $g_{p_1}, g_{p_2}, g_{p_3}$ of the subgroups $G_{p_1}, G_{p_2}, G_{p_3}$ respectively. Every element of G can then be expressed as $g_{p_1}^{a_1} g_{p_2}^{a_2} g_{p_3}^{a_3}$ for some values of a_1, a_2, a_3. We denote an element of G by (a_1, a_2, a_3). The element $e(g_{p_1}, g_{p_1})^{a_1} e(g_{p_2}, g_{p_2})^{a_2} e(g_{p_3}, g_{p_3})^{a_3}$ in G_T will be denoted by $[a_1, a_2, a_3]$. We use capital letters to denote random variables, and we reuse random variables to denote relationships between elements. For example, $X = (X_1, Y_1, Z_1)$ is a random element of G, and $Y = (X_1, Y_2, Z_2)$ is another random element that shares the same component in the G_{p_1} subgroup.

Given random variables $X, \{A_i\}$ expressed in this form, we say that X is *dependent* on $\{A_i\}$ if there exists values $\lambda_i \in \mathbb{Z}_n$ such that $X = \sum_i \lambda_i A_i$ as formal random variables. Otherwise, we say that X is *independent* of $\{A_i\}$. We note the following two theorems from [29]:

Theorem 34 *(Theorem A.1 of [29]) Let $n = \prod_{i=1}^{m} p_i$ be a product of distinct primes, each greater than 2^λ. Let $\{A_i\}$ be random variables over G, and let $\{B_i\}, T_0, T_1$ be random variables over G_T, where all random variables have degree at most t. Consider the following experiment in the generic group model:*

An algorithm is given $n, \{A_i\}$, and $\{B_i\}$. A random bit b is chosen, and the adversary is given T_b. The algorithm outputs a bits b', and succeeds if $b' = b$. The algorithm's advantage is the absolute value of the difference between its success probability and $\frac{1}{2}$.

Say each of T_0 and T_1 is independent of $\{B_i\} \cup \{e(A_i, A_j)\}$. Then given any algorithm $\mathcal{A}$ issuing at most q instructions and having advantage δ in the above experiment, $\mathcal{A}$ can be used to find a nontrivial factor of n (in time polynomial in λ and the running time of $\mathcal{A}$) with probability at least $\delta - \mathcal{O}(q^2 t / 2^\lambda)$.

Theorem 35 *(Theorem A.2 of [29]) Let $n = \prod_{i=1}^{m} p_i$ be a product of distinct primes, each greater than 2^λ. Let $\{A_i\}, T_0, T_1$ be random variables over G, and let $\{B_i\}$ be random variables over G_T, where all random variables have degree at most t. Consider the same experiment as in the theorem above.*

Let $S := \{i | e(T_0, A_i) \neq e(T_1, A_i)\}$ (where inequality refers to inequality as formal polynomials). Say each of T_0 and T_1 is independent of $\{A_i\}$, and furthermore that for all $k \in S$ it holds that $e(T_0, A_k)$ is independent of $\{B_i\} \cup \{e(A_i, A_j)\} \cup \{e(T_0, A_i)\}_{i \neq k}$, and $e(T_1, A_k)$ is independent of $\{B_i\} \cup \{e(A_i, A_j)\} \cup \{e(T_1, A_i)\}_{i \neq k}$. Then given any algorithm $\mathcal{A}$ issuing at most q instructions and having advantage δ, the algorithm can be used to find a nontrivial factor of n (in time polynomial in λ and the running time of $\mathcal{A}$) with probability at least $\delta - \mathcal{O}(q^2 t / 2^\lambda)$.

We apply these theorems to prove the security of our assumptions in the generic group model.

Assumption 1 We apply Theorem 35. We can express this assumption as:

$$A_1 = (1, 0, 0), A_2 = (0, 0, 1),$$

$T_0 = (X_1, X_2, 0), T_1 = (X_1, 0, 0)$.

We note that $S = \emptyset$ in this case. It is clear that T_0 and T_1 are both independent of $\{A_1, A_2\}$ because X_1 does not appear in A_1 or A_2. Thus, Assumption 1 is generically secure, assuming it is hard to find a nontrivial factor of n.

Assumption 2 We apply Theorem 35. We can express this assumption as:

$$A_1 = (1, 0, 0), A_2 = (X_1, 1, 0), A_3 = (Y_1, 0, 0), A_4 = (0, X_2, 1),$$

$$T_0 = (Z_1, Z_2, Z_3), T_1 = (Z_1, 0, Z_3).$$

We note that $S = \{2, 4\}$ in this case. It is clear that T_0 and T_1 are both independent of $\{A_i\}$ since Z_1 does not appear in the A_i's, for example. We see that $e(T_0, A_2)$ is independent of $\{e(A_i, A_j)\} \cup \{e(T_0, A_i)\}_{i \neq 2}$ because it is impossible to obtain $X_1 Z_1$ in the first coordinate of a combination of elements of $\{e(A_i, A_j)\} \cup \{e(T_0, A_i)\}_{i \neq 2}$. This also allows us to conclude that $e(T_1, A_2)$ is independent of $\{e(A_i, A_j)\} \cup \{e(T_1, A_i)\}_{i \neq 2}$. We similarly note that $e(T_0, A_4)$ is independent of $\{e(A_i, A_j)\} \cup \{e(T_0, A_i)\}_{i \neq 4}$ and $e(T_1, A_4)$ is independent of $\{e(A_i, A_j)\} \cup \{e(T_1, A_i)\}_{i \neq 4}$ because we cannot obtain Z_3 in the third coordinate. Thus, Assumption 2 is generically secure, assuming it is hard to find a nontrivial factor of n.

Assumption 3 We apply Theorem 34. We can express this assumption as:

$$A_1 = (1, 0, 0), A_2 = (B, 1, 0), A_3 = (0, 0, 1), A_4 = (S, X_2, 0), A_5 = (0, Y_2, 0),$$

$$T_0 = [BS, 0, 0], T_2 = [Z_1, Z_2, Z_3].$$

T_1 is independent of $\{e(A_i, A_j)\}$ because Z_1, Z_2, Z_3 do not appear in $\{A_i\}$. T_0 is independent of $\{e(A_i, A_j)\}$ because the only way to obtain BS in the first coordinate is to take $e(A_2, A_4)$, but then we are left with an X_2 in the second coordinate that cannot be canceled. Thus, Assumption 3 is generically secure, assuming it is hard to find a nontrivial factor of n.

B Appendices for Predicate Encryption

B.1 Dual Pairing Vector Spaces by Direct Product of Asymmetric Pairing Groups

Definition 36 *"Asymmetric bilinear pairing groups"* $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ *are a tuple of a prime* q*, cyclic (multiplicative) groups* $\mathbb{G}_1, \mathbb{G}_2$ *and* $\mathbb{G}_T$ *of order* q*,* $g_1 \neq 1 \in \mathbb{G}_1, g_2 \neq 1 \in \mathbb{G}_2$*, and a polynomial-time computable nondegenerate bilinear pairing* $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ *i.e.,* $e(g_1^s, g_2^t) = e(g_1, g_2)^{st}$ *and* $e(g_1, g_2) \neq 1$.

Let $\mathcal{G}_{\mathsf{bpg}}$ *be an algorithm that takes input* 1^λ *and outputs a description of bilinear pairing groups* $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ *with security parameter* λ.

Definition 37 *"Dual pairing vector spaces (DPVS)"* $(q, \mathbb{V}, \mathbb{V}^*, \mathbb{G}_T, \mathbb{A}, \mathbb{A}^*, e)$ *by direct product of asymmetric pairing groups* $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ *are a tuple of a prime* q*, two* N*-dimensional vector spaces* $\mathbb{V} := \overbrace{\mathbb{G}_1 \times \cdots \times \mathbb{G}_1}^{N}$ *and* $\mathbb{V}^* := \overbrace{\mathbb{G}_2 \times \cdots \times \mathbb{G}_2}^{N}$ *over* $\mathbb{F}_q$*, a cyclic group* $\mathbb{G}_T$ *of order* q*, and their canonical bases i.e.,* $\mathbb{A} := (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N)$ *of* $\mathbb{V}$ *and* $\mathbb{A}^* := (\boldsymbol{a}_1^*, \ldots, \boldsymbol{a}_N^*)$ *of* $\mathbb{V}^*$*, where* $\boldsymbol{a}_i := (\overbrace{1, \ldots, 1}^{i-1}, g_1, \overbrace{1, \ldots, 1}^{N-i})$ *and* $\boldsymbol{a}_i^* := (\overbrace{1, \ldots, 1}^{i-1}, g_2, \overbrace{1, \ldots, 1}^{N-i})$*, and pairing* $e : \mathbb{V} \times \mathbb{V}^* \to \mathbb{G}_T$.

The pairing is defined by $e(\boldsymbol{x},\boldsymbol{y}) := \prod_{i=1}^{N} e(d_i,h_i) \in \mathbb{G}_T$ *where* $\boldsymbol{x} := (d_1,\ldots,d_N) \in \mathbb{V}$ *and* $\boldsymbol{y} := (h_1,\ldots,h_N) \in \mathbb{V}^*$. *This is nondegenerate bilinear i.e.,* $e(s\boldsymbol{x},t\boldsymbol{y}) = e(\boldsymbol{x},\boldsymbol{y})^{st}$ *and if* $e(\boldsymbol{x},\boldsymbol{y}) = 1$ *for all* $\boldsymbol{y} \in \mathbb{V}$, *then* $\boldsymbol{x} = \boldsymbol{0}$. *For all* i *and* j, $e(\boldsymbol{a}_i,\boldsymbol{a}_j^*) = g_T^{\delta_{i,j}}$ *where* $\delta_{i,j} = 1$ *if* $i = j$, *and* 0 *otherwise, and* $g_T := e(g_1,g_2) \neq 1 \in \mathbb{G}_T$.

DPVS also has linear transformations $\phi_{i,j}$ *on* $\mathbb{V}$ *s.t.* $\phi_{i,j}(\boldsymbol{a}_j) = \boldsymbol{a}_i$ *and* $\phi_{i,j}(\boldsymbol{a}_k) = \boldsymbol{0}$ *if* $k \neq j$, *which can be easily achieved by* $\phi_{i,j}(\boldsymbol{x}) := (\overbrace{1,\ldots,1}^{i-1},d_j,\overbrace{1,\ldots,1}^{N-i})$ *where* $\boldsymbol{x} := (d_1,\ldots,d_N)$. *Moreover, linear transformation* $\phi_{i,j}^*$ *on* $\mathbb{V}^*$ *s.t.* $\phi_{i,j}^*(\boldsymbol{a}_j^*) = \boldsymbol{a}_i^*$ *and* $\phi_{i,j}^*(\boldsymbol{a}_k^*) = \boldsymbol{0}$ *if* $k \neq j$ *can be easily achieved by* $\phi_{i,j}^*(\boldsymbol{y}) := (\overbrace{1,\ldots,1}^{i-1},h_j,\overbrace{1,\ldots,1}^{N-i})$ *where* $\boldsymbol{y} := (h_1,\ldots,h_N)$. *We call* $\phi_{i,j}$ *and* $\phi_{i,j}^*$ *"distortion maps".*

B.2 Proof of Lemma 15

As in Appendix A.2, using a generator g, every element of $\mathbb{G}$ can be expressed as g^a for some value of a. We denote an element of $\mathbb{G}$ shortly by a, and we use capital letters to denote (formal) random variables.

Let $\mathcal{Q} := \{\ g, g^{\kappa}, g^{\omega+\gamma_i h_i}, g^{\gamma_i}, g^{h_i}\ (1 \le i \le n),\ g^{\gamma_i h_j}\ (1 \le i \neq j \le n)\ \in \mathbb{G}\ \}$, i.e., the elements in the n-eDDH instance except for $Y_\beta \in \mathbb{G}$, and $\mathcal{P} := \{P_i\} := \{\ 1, K, \Omega + \Gamma_i H_i, \Gamma_i, H_i\ (1 \le i \le n), \Gamma_i H_j\ (1 \le i \neq j \le n)\ \}$, the set of polynomials of formal random variables obtained from $\mathcal{Q}$. It is clear that $\mathcal{P}$ and the target $K\Omega$ are linearly independent over $\mathbb{F}_q$ (as formal polynomials). Therefore, Lemma 15 follows from Theorem A.2 in [8] and Lemma 38 below.

Lemma 38 *If*

$$\sum_i a_i \cdot K\Omega P_i = \sum_{j,k} b_{j,k} \cdot P_j P_k \tag{1}$$

holds among elements $P_i \in \mathcal{P}$ *and* $a_i, b_{j,k} \in \mathbb{F}_q$, *then all* $a_i = 0$, *i.e.,* $\{K\Omega P_i\}$ *are independent of* $\{P_j P_k\}$.

Proof. The set $\{K\Omega P \mid P \in \mathcal{P}\}$ is given as

$$\{K\Omega,\ K^2\Omega,\ K\Omega(\Omega+\Gamma_i H_i),\ K\Omega\Gamma_i,\ K\Omega H_i\ (1 \le i \le n),\quad K\Omega\Gamma_i H_j\ \ (1 \le i \neq j \le n)\}.$$

By the following facts, we conclude that no terms in the above set appear in the linear relation (1), i.e., all $a_i = 0$.

- As for $K\Omega(\Omega + \Gamma_i H_i)$, there exists only one way, $(\Omega + \Gamma_i H_i)^2$, to obtain Ω^2 from a product of two elements in $\mathcal{P}$. However, obviously, since $(\Omega + \Gamma_i H_i)^2$ does not contain K, $K\Omega(\Omega + \Gamma_i H_i)$ cannot be included in relation (1).
- Since neither $K^2, K\Gamma_i, KH_i$, nor $K\Gamma_i H_j$ is included in $\mathcal{P}$, element $K^2\Omega, K\Omega\Gamma_i, K\Omega$, or $K\Omega\Gamma_i H_j$ cannot be obtained from a product of two elements in $\mathcal{P}$. Therefore, any of these cannot be included in linear relation (1).
- To obtain $K\Omega$, we must use the product of K and $\Omega + \Gamma_i H_i$. Then, we need to obtain $K\Gamma_i H_i$ from a product of two elements in $\mathcal{P}$. However, since each term containing Γ_i except for $\Omega + \Gamma_i H_i$ contains some H_j $(j \neq i)$, element $K\Gamma_i H_i$ cannot be obtained from the desired product. Therefore, $K\Omega$ cannot be included in linear relation (1).

□

B.3 Proofs of Lemmas 20 and 22

B.3.1 Preliminary Lemma for Lemmas 42 and 45

We will use the following lemma in the proofs of Lemmas 42 and 45.

Lemma 39 *Let $(q, \mathbb{V}, \mathbb{G}_T, \mathbb{A}, e)$ be dual pairing vector spaces by direct product of symmetric pairing groups. Using $\{\phi_{i,j}\}$, we can efficiently sample a random linear transformation*

$$W := \sum_{i=1,j=1}^{N,N} r_{i,j}\phi_{i,j}$$

of $\mathbb{V}$ with random coefficients $\{r_{i,j}\}_{i,j\in\{1,\dots,N\}} \xleftarrow{\mathsf{U}} \mathbb{F}_q^{N\times N}$. At that time, the matrix $(r^*_{i,j}) := (\{r_{i,j}\}^{-1})^T$ defines the adjoint action on $\mathbb{V}$ for pairing e, i.e., $e(W(\boldsymbol{x}), (W^{-1})^T(\boldsymbol{y})) = e(\boldsymbol{x}, \boldsymbol{y})$ for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{V}$, via

$$(W^{-1})^T := \sum_{i=1,j=1}^{N,N} r^*_{i,j}\phi_{i,j}.$$

B.3.2 Proof of Lemma 20

To prove Lemma 20, we first define a simplified version of Problem 1.

Definition 40 *Simple Problem 1 is to guess $\beta \in \{0,1\}$, given $(\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \{\boldsymbol{e}_{\beta,i}\}_{i=1,\dots,n}) \xleftarrow{\mathsf{R}} \mathcal{G}^{\mathsf{SP1}}_\beta(1^\lambda, n)$, where*

$$\begin{aligned}
&\mathcal{G}^{\mathsf{SP1}}_\beta(1^\lambda, n): \ (\mathsf{param}_{\mathbb{V}}, \mathbb{B}, \mathbb{B}^*) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^\lambda, 2n+3),\\
&\quad \widehat{\mathbb{B}} := (\boldsymbol{b}_1, \dots, \boldsymbol{b}_n, \boldsymbol{b}_{2n+1}, \boldsymbol{b}_{2n+3}), \quad \widehat{\mathbb{B}}^* := (\boldsymbol{b}^*_1, \dots, \boldsymbol{b}^*_n, \boldsymbol{b}^*_{2n+1}, \boldsymbol{b}^*_{2n+2}),\\
&\quad \delta_1, \delta_{2,i} \xleftarrow{\mathsf{U}} \mathbb{F}_q, \quad \rho \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times \quad \text{for } i = 1, \dots, n,\\
&\quad \text{for } i = 1, \dots, n,\\
&\qquad \boldsymbol{e}_{0,i} := \delta_1 \boldsymbol{b}_i + \delta_{2,i}\boldsymbol{b}_{2n+3},\\
&\qquad \boldsymbol{e}_{1,i} := \delta_1 \boldsymbol{b}_i + \rho \boldsymbol{b}_{n+i} + \delta_{2,i}\boldsymbol{b}_{2n+3},\\
&\quad \text{return} \ \ (\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \{\boldsymbol{e}_{\beta,i}\}_{i=1,\dots,n}),
\end{aligned}$$

for $\beta \xleftarrow{\mathsf{U}} \{0,1\}$. For a probabilistic machine $\mathcal{D}$, we define the advantage of adversary $\mathcal{D}$ for Simple Problem 1 as:

$$\mathsf{Adv}^{\mathsf{SP1}}_{\mathcal{D}}(\lambda) := \left|\mathsf{Pr}\left[\mathcal{D}(1^\lambda, \varrho) \to 1 \,\middle|\, \varrho \xleftarrow{\mathsf{R}} \mathcal{G}^{\mathsf{SP1}}_0(1^\lambda, n)\right] - \mathsf{Pr}\left[\mathcal{D}(1^\lambda, \varrho) \to 1 \,\middle|\, \varrho \xleftarrow{\mathsf{R}} \mathcal{G}^{\mathsf{SP1}}_1(1^\lambda, n)\right]\right|.$$

Simple Problem 1 is reduced to Problem 1 (Lemma 41), and the n-eDDH problem is reduced to Simple Problem 1 (Lemma 42). Therefore, we obtain Lemma 20 by combining Lemmas 41 and 42.

Lemma 41 *The output distribution of Simple Problem 1, $(\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \{\boldsymbol{e}_{\beta,i}\}_{i=1,\dots,n})$, is exactly the same as that of Problem 1. In particular, for any adversary $\mathcal{B}$, there is a probabilistic machine $\mathcal{D}$, whose running time is the same as that of $\mathcal{B}$, such that for any security parameter λ, $\mathsf{Adv}^{\mathsf{SP1}}_{\mathcal{D}}(\lambda) = \mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}}(\lambda)$.*

Proof. Given a Simple Problem 1 instance, $(\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \{\boldsymbol{e}_{\beta,i}\}_{i=1,\dots,n})$, we generate $U := (u_{i,j}) \xleftarrow{\mathsf{U}} GL(n, \mathbb{F}_q)$, and calculate $Z := (z_{i,j}) := (U^{\mathrm{T}})^{-1}$.

We then calculate $\{\boldsymbol{d}_{n+1},\ldots,\boldsymbol{d}_{2n}\}$ from $\{\boldsymbol{b}_{n+1},\ldots,\boldsymbol{b}_{2n}\}$ as

$$\boldsymbol{d}_{n+j} := \textstyle\sum_{i=1}^{n} z_{i,j}\boldsymbol{b}_{n+i} \quad \text{for } j = 1,\ldots,n.$$

Then,

$$\boldsymbol{b}_{n+i} = \textstyle\sum_{j=1}^{n} u_{i,j}\boldsymbol{d}_{n+j} \quad \text{for } i = 1,\ldots,n$$

since $UZ^{\mathrm{T}} = I_n$, and, for $i = 1,\ldots,n$,

$$\begin{aligned}\boldsymbol{e}_{0,i} &= \delta_1\boldsymbol{b}_i + \delta_{2,i}\boldsymbol{b}_{2n+3},\\ \boldsymbol{e}_{1,i} &= \delta_1\boldsymbol{b}_i + \rho\boldsymbol{b}_{n+i} + \delta_{2,i}\boldsymbol{b}_{2n+3} = \delta_1\boldsymbol{b}_i + \rho\textstyle\sum_{j=1}^{n} u_{i,j}\boldsymbol{d}_{n+j} + \delta_{2,i}\boldsymbol{b}_{2n+3}.\end{aligned}$$

Let basis $\mathbb{D} := (\boldsymbol{b}_1,\ldots,\boldsymbol{b}_n,\boldsymbol{d}_{n+1},\ldots,\boldsymbol{d}_{2n},\boldsymbol{b}_{2n+1},\boldsymbol{b}_{2n+2},\boldsymbol{b}_{2n+3})$. Both bases $\mathbb{B}$ and $\mathbb{D}$ are consistent with $(\mathsf{param}_{\mathbb{V}},\widehat{\mathbb{B}},\widehat{\mathbb{B}}^*)$. Hence, we can consider the distribution of $(\mathsf{param}_{\mathbb{V}},\widehat{\mathbb{B}},\widehat{\mathbb{B}}^*,\{\boldsymbol{e}_{\beta,i}\}_{i=1,\ldots,n})$ as $\left\{\varrho \;\middle|\; \varrho \xleftarrow{\mathsf{R}} \mathcal{G}_\beta^{\mathsf{P1}}(1^\lambda,n)\right\}$ by using basis $\mathbb{D}$.

The other direction can be shown exactly in the same manner. □

Lemma 42 *For any adversary $\mathcal{D}$, there is a probabilistic machine $\mathcal{C}$, whose running time is essentially the same as that of $\mathcal{D}$, such that for any security parameter λ, $\mathsf{Adv}^{n\text{-eDDH}}_{\mathcal{C}}(\lambda) = \mathsf{Adv}^{\mathsf{SP1}}_{\mathcal{D}}(\lambda)$.*

Proof. Given an n-eDDH instance

$$\mathsf{param}_{\mathbb{G}},\ g, g^{\kappa}, \{g^{\omega+\gamma_i h_i}, g^{\gamma_i}, g^{h_i}\}_{1\le i\le n},\ \{g^{\gamma_i h_j}\}_{1\le i\neq j\le n},\ Y_\beta,$$

$\mathcal{C}$ sets $(2n+3)\times(2n+3)$ matrices

$$\Pi := (\pi_{i,j}) := \begin{pmatrix} \kappa\cdot I_n & I_n & & & \\ -\kappa\cdot I_n & 0_n & & & \\ & & 1 & & \\ & & & 1 & \\ & \overrightarrow{h} & & & 1 \end{pmatrix},$$

$$\Pi^* := (\pi^*_{i,j}) := \begin{pmatrix} 0_n & I_n & & & -\overrightarrow{h}^{\mathrm{T}} \\ -\kappa^{-1}\cdot I_n & I_n & & & -\overrightarrow{h}^{\mathrm{T}} \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{pmatrix},$$

where $\overrightarrow{h} := (h_1,\ldots,h_n)$. Then, $\Pi\cdot(\Pi^*)^{\mathrm{T}} = I_{2n+3}$.

$\mathcal{C}$ can compute $\boldsymbol{u}_i := \sum_{j=1}^{2n+3}\pi_{i,j}\boldsymbol{a}_j$ $(i = 1,\ldots,2n+3)$ and $\boldsymbol{u}^*_i := \sum_{j=1}^{2n+3}\pi^*_{i,j}\boldsymbol{a}_j$ $(i = 1,\ldots,n,2n+1,2n+2)$ from the above n-eDDH instance.

$\mathcal{C}$ then sets

$$\begin{aligned}\boldsymbol{v}_{\beta,i} := (&\overbrace{1,\ldots,1}^{i-1}, Y_\beta, \overbrace{1,\ldots,1}^{n-i},\\ &g^{h_1\gamma_i},\ \ldots,\ g^{h_{i-1}\gamma_i},\ g^{\omega+h_i\gamma_i},\ g^{h_{i+1}\gamma_i},\ \ldots,\ g^{h_n\gamma_i},\ 1, 1,\ g^{\gamma_i}).\end{aligned}$$

$\mathcal{C}$ generates random linear transformation W given in Lemma 39, then sets

$$\begin{aligned}
&\boldsymbol{b}_i := W(\boldsymbol{u}_i) \ \text{ for } i = 1, \ldots, n, 2n+1, 2n+3,\\
&\boldsymbol{b}^*_i := (W^{-1})^{\mathrm{T}}(\boldsymbol{u}^*_i) \ \text{ for } i = 1, \ldots, n, 2n+1, 2n+2,\\
&\widehat{\mathbb{B}} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n, \boldsymbol{b}_{2n+1}, \boldsymbol{b}_{2n+3}), \quad \widehat{\mathbb{B}}^* := (\boldsymbol{b}^*_1, \ldots, \boldsymbol{b}^*_n, \boldsymbol{b}^*_{2n+1}, \boldsymbol{b}^*_{2n+2}),\\
&\boldsymbol{e}_{\beta,i} := W(\boldsymbol{v}_{\beta,i}) \ \text{ for } i = 1, \ldots, n.
\end{aligned}$$

$\mathcal{C}$ then gives $(\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \{\boldsymbol{e}_{\beta,i}\}_{i=1,\ldots,n})$ to $\mathcal{D}$, and outputs $\beta' \in \{0,1\}$ if $\mathcal{D}$ outputs β'.

If we set $\delta_1 := \omega,\ \delta_{2,i} := \gamma_i$, then

$$\begin{aligned}
\boldsymbol{v}_{\beta,i} = (&\overbrace{1, \ldots, 1}^{i-1}, Y_\beta, \overbrace{1, \ldots, 1}^{n-i},\\
&g^{h_1\delta_{2,i}}, \ \ldots, \ g^{h_{i-1}\delta_{2,i}}, \ g^{\delta_1 + h_i\delta_{2,i}}, \ g^{h_{i+1}\delta_{2,i}}, \ \ldots, \ g^{h_n\delta_{2,i}}, \ 1, 1, \ g^{\delta_{2,i}}).
\end{aligned}$$

If $\beta = 0$, i.e., $Y_\beta = g^{\kappa\omega}$ $(= g^{\delta_1\kappa})$, then

$$\boldsymbol{e}_{\beta,i} = \delta_1 \boldsymbol{b}_i + \delta_{2,i} \boldsymbol{b}_{2n+3},$$

and the distribution of $(\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \{\boldsymbol{e}_{\beta,i}\}_{i=1,\ldots,n})$ is exactly the same as $\left\{ \varrho \;\middle|\; \varrho \xleftarrow{\mathsf{R}} \mathcal{G}_0^{\mathsf{SP1}}(1^\lambda, n) \right\}$.

If $\beta = 1$, i.e., Y_β $(= g^\psi)$ is uniformly distributed in $\mathbb{G}$, then

$$\boldsymbol{e}_{\beta,i} = \delta_1 \boldsymbol{b}_i + \rho \boldsymbol{b}_{n+i} + \delta_{2,i} \boldsymbol{b}_{2n+3},$$

where $\rho := \delta_1 - \kappa^{-1}\psi$, and ρ is also uniformly distributed. Therefore, the distribution of $(\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \{\boldsymbol{e}_{\beta,i}\}_{i=1,\ldots,n})$ is exactly the same as $\left\{ \varrho \;\middle|\; \varrho \xleftarrow{\mathsf{R}} \mathcal{G}_1^{\mathsf{SP1}}(1^\lambda, n) \right\}$. $\square$

B.3.3 Proof of Lemma 22

To prove Lemma 22, we first define a simplified version of Problem 2.

Definition 43 *Simple Problem 2 is to guess $\beta \in \{0,1\}$, given $(\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \{\boldsymbol{h}^*_{\beta,i}, \boldsymbol{e}_i\}_{i=1,\ldots,n}) \xleftarrow{\mathsf{R}} \mathcal{G}_\beta^{\mathsf{SP2}}(1^\lambda, n)$, where*

$$\begin{aligned}
&\mathcal{G}_\beta^{\mathsf{SP2}}(1^\lambda, n): \ (\mathsf{param}_{\mathbb{V}}, \mathbb{B}, \mathbb{B}^*) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^\lambda, 2n+3),\\
&\quad \widehat{\mathbb{B}} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n, \boldsymbol{b}_{2n+1}, \boldsymbol{b}_{2n+3}), \quad \widehat{\mathbb{B}}^* := (\boldsymbol{b}^*_1, \ldots, \boldsymbol{b}^*_{2n+2}),\\
&\quad \omega, \gamma_i, \delta \xleftarrow{\mathsf{U}} \mathbb{F}_q, \ \rho, \tau \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times \text{ for } i = 1, \ldots, n,\\
&\quad \text{for } i = 1, \ldots, n,\\
&\qquad \boldsymbol{h}^*_{0,i} := \omega \boldsymbol{b}^*_i + \gamma_i \boldsymbol{b}^*_{2n+2},\\
&\qquad \boldsymbol{h}^*_{1,i} := \omega \boldsymbol{b}^*_i + \tau \boldsymbol{b}^*_{n+i} + \gamma_i \boldsymbol{b}^*_{2n+2},\\
&\qquad \boldsymbol{e}_i := \delta \boldsymbol{b}_i + \rho \boldsymbol{b}_{n+i},\\
&\quad \text{return } \ (\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \{\boldsymbol{h}^*_{\beta,i}, \boldsymbol{e}_i\}_{i=1,\ldots,n}),
\end{aligned}$$

for $\beta \xleftarrow{\mathsf{U}} \{0,1\}$. For a probabilistic machine $\mathcal{D}$, we define the advantage of adversary $\mathcal{D}$ for Simple Problem 2 as:

$$\mathsf{Adv}_{\mathcal{D}}^{\mathsf{SP2}}(\lambda) := \left| \mathsf{Pr}\left[\mathcal{D}(1^\lambda, \varrho) \to 1 \;\middle|\; \varrho \xleftarrow{\mathsf{R}} \mathcal{G}_0^{\mathsf{SP2}}(1^\lambda, n) \right] - \mathsf{Pr}\left[\mathcal{D}(1^\lambda, \varrho) \to 1 \;\middle|\; \varrho \xleftarrow{\mathsf{R}} \mathcal{G}_1^{\mathsf{SP2}}(1^\lambda, n) \right] \right|.$$

Simple Problem 2 is reduced to Problem 2 (Lemma 44), and the n-eDDH problem is reduced to Simple Problem 2 (Lemma 45). Therefore, we obtain Lemma 22 by combining Lemmas 44 and 45.

Lemma 44 *For any adversary $\mathcal{B}$, there is a probabilistic machine $\mathcal{D}$, whose running time is essentially the same as that of $\mathcal{B}$, such that for any security parameter λ, $\mathsf{Adv}^{\mathsf{SP2}}_{\mathcal{D}}(\lambda) = \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}}(\lambda)$.*

Proof. Given a Simple Problem 2 instance, $(\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \{\boldsymbol{h}^*_{\beta,i}, \boldsymbol{e}_i\}_{i=1,\ldots,n})$, $\mathcal{D}$ generates $U := (u_{i,j}) \xleftarrow{\mathsf{U}} GL(n, \mathbb{F}_q)$, and calculates $Z := (z_{i,j}) := (U^{\mathrm{T}})^{-1}$.

$\mathcal{D}$ then calculates $\{\boldsymbol{d}_{n+1}, \ldots, \boldsymbol{d}_{2n}\}$ and $\{\boldsymbol{d}^*_{n+1}, \ldots, \boldsymbol{d}^*_{2n}\}$ from $\{\boldsymbol{b}_{n+1}, \ldots, \boldsymbol{b}_{2n}\}$ and $\{\boldsymbol{b}^*_{n+1}, \ldots, \boldsymbol{b}^*_{2n}\}$, respectively, as

$$\boldsymbol{d}_{n+j} := \textstyle\sum_{i=1}^{n} z_{i,j} \boldsymbol{b}_{n+i}, \quad \boldsymbol{d}^*_{n+j} := \textstyle\sum_{i=1}^{n} u_{i,j} \boldsymbol{b}^*_{n+i} \quad \text{for } j = 1, \ldots, n.$$

Then,

$$\boldsymbol{b}_{n+i} = \textstyle\sum_{j=1}^{n} u_{i,j} \boldsymbol{d}_{n+j}, \quad \boldsymbol{b}^*_{n+i} = \textstyle\sum_{j=1}^{n} z_{i,j} \boldsymbol{d}^*_{n+j} \quad \text{for } i = 1, \ldots, n,$$

since $UZ^{\mathrm{T}} = I_n$, and, for $i = 1, \ldots, n$,

$$\begin{aligned}
&\boldsymbol{h}^*_{0,i} = \omega \boldsymbol{b}^*_i + \gamma_i \boldsymbol{b}^*_{2n+2}, \\
&\boldsymbol{h}^*_{1,i} = \omega \boldsymbol{b}^*_i + \tau \boldsymbol{b}^*_{n+i} + \gamma_i \boldsymbol{b}^*_{2n+2} = \omega \boldsymbol{b}^*_i + \tau \textstyle\sum_{j=1}^{n} z_{i,j} \boldsymbol{d}^*_{n+j} + \gamma_i \boldsymbol{b}^*_{2n+2}, \\
&\boldsymbol{e}_i = \delta \boldsymbol{b}_i + \rho \boldsymbol{b}_{n+i} = \delta \boldsymbol{b}_i + \rho \textstyle\sum_{j=1}^{n} u_{i,j} \boldsymbol{d}_{n+j}.
\end{aligned}$$

Let

$$\widehat{\mathbb{D}}^* := (\boldsymbol{b}^*_1, \ldots, \boldsymbol{b}^*_n, \boldsymbol{d}^*_{n+1}, \ldots, \boldsymbol{d}^*_{2n}, \boldsymbol{b}^*_{2n+1}, \boldsymbol{b}^*_{2n+2}).$$

$\mathcal{D}$ then gives $(\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{D}}^*, \{\boldsymbol{h}^*_{\beta,i}, \boldsymbol{e}_i\}_{i=1,\ldots,n})$ to $\mathcal{B}$, and outputs $\beta' \in \{0,1\}$ if $\mathcal{B}$ outputs β'.

By the construction, the distribution of $(\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{D}}^*, \{\boldsymbol{h}^*_{\beta,i}, \boldsymbol{e}_i\}_{i=1,\ldots,n})$, is exactly the same as $\left\{ \varrho \;\middle|\; \varrho \xleftarrow{\mathsf{R}} \mathcal{G}^{\mathsf{P2}}_{\beta}(1^{\lambda}, n) \right\}$. $\square$

Lemma 45 *For any adversary $\mathcal{D}$, there is a probabilistic machine $\mathcal{C}$, whose running time is essentially the same as that of $\mathcal{D}$, such that for any security parameter λ, $\mathsf{Adv}^{n\text{-}\mathsf{eDDH}}_{\mathcal{C}}(\lambda) = \mathsf{Adv}^{\mathsf{SP2}}_{\mathcal{D}}(\lambda)$.*

Proof. Given an n-eDDH instance

$$\mathsf{param}_{\mathbb{G}},\ g, g^{\kappa}, \{g^{\omega+\gamma_i h_i}, g^{\gamma_i}, g^{h_i}\}_{1 \le i \le n},\ \{g^{\gamma_i h_j}\}_{1 \le i \ne j \le n},\ Y_{\beta},$$

$\mathcal{C}$ sets $(2n+3)\times(2n+3)$ matrices

$$\Pi^* := (\pi^*_{i,j}) := \begin{pmatrix} \kappa\cdot I_n & I_n & & \\ -\kappa\cdot I_n & 0_n & & \\ & & 1 & \\ & \overrightarrow{h} & & 1 \\ & & & & 1 \end{pmatrix},$$

$$\Pi := (\pi_{i,j}) := \begin{pmatrix} 0_n & I_n & -\overrightarrow{h}^{\mathrm{T}} & \\ -\kappa^{-1}\cdot I_n & I_n & -\overrightarrow{h}^{\mathrm{T}} & \\ & & 1 & \\ & & & 1 \\ & & & & 1 \end{pmatrix},$$

where $\overrightarrow{h} := (h_1,\ldots,h_n)$. Then, $\Pi\cdot(\Pi^*)^{\mathrm{T}} = I_{2n+3}$.

$\mathcal{C}$ can compute $\boldsymbol{u}_i := \sum_{j=1}^{2n+3}\pi_{i,j}\boldsymbol{a}_j$ $(i=1,\ldots,n,2n+1,2n+3)$ and $\boldsymbol{u}^*_i := \sum_{j=1}^{2n+3}\pi^*_{i,j}\boldsymbol{a}_j$ $(i=1,\ldots,2n+2)$ from the above n-eDDH instance.

$\mathcal{C}$ then generates $\xi,\varphi \xleftarrow{\cup} \mathbb{F}_q$, and

$$\boldsymbol{v}_i := (\overbrace{1,\ldots,1}^{i-1}, g^{\xi}, \overbrace{1,\ldots,1}^{n-1}, g^{\varphi}, \overbrace{1,\ldots,1}^{n-i+1}, g^{-\varphi h_i}, 1),$$
$$\boldsymbol{w}^*_{\beta,i} := (\overbrace{1,\ldots,1}^{i-1}, Y_\beta, \overbrace{1,\ldots,1}^{n-i}, g^{\gamma_i h_1},\ldots,g^{\gamma_i h_{i-1}}, g^{\omega+\gamma_i h_i}, g^{\gamma_i h_{i+1}},\ldots,g^{\gamma_i h_n}, 1, g^{\gamma_i}, 1).$$

$\mathcal{C}$ generates random linear transformation W given in Lemma 39, then sets

$$\begin{aligned} &\boldsymbol{b}_i := W(\boldsymbol{u}_i) \ \text{ for } i=1,\ldots,n,2n+1,2n+3,\\ &\boldsymbol{b}^*_i := (W^{-1})^{\mathrm{T}}(\boldsymbol{u}^*_i) \ \text{ for } i=1,\ldots,2n+2,\\ &\widehat{\mathbb{B}} := (\boldsymbol{b}_1,\ldots,\boldsymbol{b}_n,\boldsymbol{b}_{2n+1},\boldsymbol{b}_{2n+3}), \quad \widehat{\mathbb{B}}^* := (\boldsymbol{b}^*_1,\ldots,\boldsymbol{b}^*_{2n+2}),\\ &\boldsymbol{e}_i := W(\boldsymbol{v}_i), \quad \boldsymbol{h}^*_{\beta,i} := (W^{-1})^{\mathrm{T}}(\boldsymbol{w}^*_{\beta,i}) \ \text{ for } i=1,\ldots,n. \end{aligned}$$

$\mathcal{C}$ then gives $(\mathsf{param}_{\mathbb{V}},\widehat{\mathbb{B}},\widehat{\mathbb{B}}^*,\{\boldsymbol{h}^*_{\beta,i},\boldsymbol{e}_i\}_{i=1,\ldots,n})$ to $\mathcal{D}$, and outputs $\beta'\in\{0,1\}$ if $\mathcal{D}$ outputs β'.

If we set

$$\rho := -\kappa\xi, \quad \delta_1 := \varphi-\rho,$$

then

$$\boldsymbol{v}_i = (\overbrace{1,\ldots,1}^{i-1}, g^{-\rho\kappa^{-1}}, \overbrace{1,\ldots,1}^{n-1}, g^{\delta_1+\rho}, \overbrace{1,\ldots,1}^{n-i+1}, g^{-(\delta_1+\rho)h_i}, 1),$$

and we see that

$$\boldsymbol{e}_i = \delta_1\boldsymbol{b}_i + \rho\boldsymbol{b}_{n+i}.$$

If $\beta=0$, i.e., $Y_\beta = g^{\kappa\omega}$, then

$$\boldsymbol{h}^*_{\beta,i} = \omega\boldsymbol{b}^*_i + \gamma_i\boldsymbol{b}^*_{2n+2},$$

and the distribution of $(\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \{\boldsymbol{h}^*_{\beta,i}, \boldsymbol{e}_i\}_{i=1,\ldots,n})$ is exactly the same as $\left\{\varrho \;\middle|\; \varrho \xleftarrow{\mathsf{R}} \mathcal{G}_0^{\mathsf{SP2}}(1^\lambda, n)\right\}$.

If $\beta = 1$, i.e., Y_β $(= g^\psi)$ is uniformly distributed in $\mathbb{G}$, then

$$\boldsymbol{h}^*_{\beta,i} = \omega \boldsymbol{b}^*_i + \tau \boldsymbol{b}^*_{n+i} + \gamma_i \boldsymbol{b}^*_{2n+2},$$

where $\tau := \omega - \kappa^{-1}\psi$, and τ is also uniformly distributed. Therefore, the distribution of $(\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \{\boldsymbol{h}^*_{\beta,i}, \boldsymbol{e}_i\}_{i=1,\ldots,n})$ is exactly the same as $\left\{\varrho \;\middle|\; \varrho \xleftarrow{\mathsf{R}} \mathcal{G}_1^{\mathsf{SP2}}(1^\lambda, n)\right\}$. □

B.4 Proof of Lemma 23

Since the inner-product of $\overrightarrow{x}(\rho U)$ and $\overrightarrow{v}(\tau Z)$ is equal to $\rho\tau(\overrightarrow{x} \cdot \overrightarrow{v})$, points on C are mapped onto C by the action of $\rho U \times \tau Z$.

For $(\overrightarrow{x}, \overrightarrow{v}) \in C$, $\overrightarrow{x} \notin \overrightarrow{v}^\perp$ (hyperplane determined by $\overrightarrow{v}$). Let $(\overrightarrow{y}_1, \ldots, \overrightarrow{y}_{n-1})$ be a basis of $\overrightarrow{v}^\perp$. Since $\overrightarrow{x} \notin \overrightarrow{v}^\perp$, the tuple $(\overrightarrow{x}, \overrightarrow{y}_1, \ldots, \overrightarrow{y}_{n-1})$ is a basis of V. Therefore, transformed vectors $(\overrightarrow{x}(\rho U), \overrightarrow{y}_1(\rho U), \ldots, \overrightarrow{y}_{n-1}(\rho U))$ is a uniformly chosen basis of V when $Z \xleftarrow{\mathsf{U}} GL(n, \mathbb{F}_q)$, $U := (Z^{-1})^{\mathrm{T}}$, and $\rho \xleftarrow{\mathsf{U}} \mathbb{F}_q^\times$.

The point $\overrightarrow{r} := \overrightarrow{v}(\tau Z)$ can be identified to the hyperplane spanned by $(\overrightarrow{y}_1(\rho U), \ldots, \overrightarrow{y}_{n-1}(\rho U))$ up to a constant multiple because of the duality. Note that $\overrightarrow{w} := \overrightarrow{x}(\rho U)$ is an independent vector outside that hyperplane. Therefore, the pair of $\overrightarrow{w}$ and $\overrightarrow{r}$ is uniformly distributed in $V \times V^*$ such that $\overrightarrow{w} \cdot \overrightarrow{r} \neq 0$, i.e., uniformly distributed in C.

B.5 Definition of Hierarchical Predicate Encryption

This section defines hierarchical predicate encryption (HPE) for the class of hierarchical inner-product predicates and its security. [1]

In a delegation system, it is required that a user who has a capability can delegate to another user a more restrictive capability. In addition to this requirement, our hierarchical inner-product encryption introduces a format of hierarchy $\overrightarrow{\mu}$ to define common delegation structure in a system.

We call a tuple of positive integers $\overrightarrow{\mu} := (n, d; \mu_1, \ldots, \mu_d)$ s.t. $\mu_0 = 0 < \mu_1 < \mu_2 < \cdots < \mu_d = n$ a format of hierarchy of depth d attribute spaces. Let Σ_ℓ $(\ell = 1, \ldots, d)$ be the sets of attributes, where each $\Sigma_\ell := \mathbb{F}_q^{\mu_\ell - \mu_{\ell-1}} \setminus \{\overrightarrow{0}\}$. Let the hierarchical attributes $\Sigma := \cup_{\ell=1}^d (\Sigma_1 \times \ldots \times \Sigma_\ell)$, where the union is a disjoint union. Then, for $\overrightarrow{v}_i \in \mathbb{F}_q^{\mu_i - \mu_{i-1}} \setminus \{\overrightarrow{0}\}$, the hierarchical predicate $f_{(\overrightarrow{v}_1, \ldots, \overrightarrow{v}_\ell)}$ on hierarchical attributes $(\overrightarrow{x}_1, \ldots, \overrightarrow{x}_h) \in \Sigma$ is defined as follows: $f_{(\overrightarrow{v}_1, \ldots, \overrightarrow{v}_\ell)}(\overrightarrow{x}_1, \ldots, \overrightarrow{x}_h) = 1$ iff $\ell \leq h$ and $\overrightarrow{x}_i \cdot \overrightarrow{v}_i = 0$ for all i s.t. $1 \leq i \leq \ell$.

Let the space of hierarchical predicates $\mathcal{F} := \{f_{(\overrightarrow{v}_1, \ldots, \overrightarrow{v}_\ell)} \mid \overrightarrow{v}_i \in \mathbb{F}_q^{\mu_i - \mu_{i-1}} \setminus \{\overrightarrow{0}\}\}$. We call h (resp. ℓ) the level of $(\overrightarrow{x}_1, \ldots, \overrightarrow{x}_h)$ (resp. $(\overrightarrow{v}_1, \ldots, \overrightarrow{v}_\ell)$).

Definition 46 *Let $\overrightarrow{\mu} := (n, d; \mu_1, \ldots, \mu_d)$ s.t. $\mu_0 = 0 < \mu_1 < \mu_2 < \cdots < \mu_d = n$ be a format of hierarchy of depth d attribute spaces. A hierarchical predicate encryption (HPE) scheme for the class of hierarchical inner-product predicates $\mathcal{F}$ over the set of hierarchical attributes Σ consists of probabilistic polynomial-time algorithms* $\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}$, *and* $\mathsf{Delegate}_\ell$ *for $\ell = 1, \ldots, d-1$. They are given as follows:*

- Setup *takes as input security parameter 1^λ and format of hierarchy $\overrightarrow{\mu}$, and outputs (master) public key* pk *and (master) secret key* sk.

[1] More general delegation structures (partial order structures) than tree hierarchical structures can be easily realized in our HPE scheme. See Appendix B.8.

- KeyGen *takes as input the master public key* pk*, secret key* sk*, and predicate vectors* $(\overrightarrow{v}_1, \ldots, \overrightarrow{v}_\ell)$*. It outputs a corresponding secret key* $\mathsf{sk}_{(\overrightarrow{v}_1,\ldots,\overrightarrow{v}_\ell)}$*.*

- Enc *takes as input the master public key* pk*, attribute vectors* $(\overrightarrow{x}_1, \ldots, \overrightarrow{x}_h)$*, where* $1 \leq h \leq d$*, and plaintext* m *in some associated plaintext space,* msg*. It returns ciphertext* c*.*

- Dec *takes as input the master public key* pk*, secret key* $\mathsf{sk}_{(\overrightarrow{v}_1,\ldots,\overrightarrow{v}_\ell)}$*, where* $1 \leq \ell \leq d$*, and ciphertext* c*. It outputs either plaintext* m *or the distinguished symbol* $\perp$*.*

- $\mathsf{Delegate}_\ell$ *takes as input the master public key* pk*,* ℓ*-th level secret key* $\mathsf{sk}_{(\overrightarrow{v}_1,\ldots,\overrightarrow{v}_\ell)}$*, and* $(\ell+1)$*-th level predicate vector* $\overrightarrow{v}_{\ell+1}$*. It returns* $(\ell+1)$*-th level secret key* $\mathsf{sk}_{(\overrightarrow{v}_1,\ldots,\overrightarrow{v}_{\ell+1})}$*.*

A HPE scheme should have the following correctness property: for all correctly generated pk and $\mathsf{sk}_{(\overrightarrow{v}_1,\ldots,\overrightarrow{v}_\ell)}$, generate $c \xleftarrow{\mathsf{R}} \mathsf{Enc}(\mathsf{pk}, m, (\overrightarrow{x}_1, \ldots, \overrightarrow{x}_h))$ and $m' := \mathsf{Dec}(\mathsf{pk}, \mathsf{sk}_{(\overrightarrow{v}_1,\ldots,\overrightarrow{v}_\ell)}, c)$. If $f_{(\overrightarrow{v}_1,\ldots,\overrightarrow{v}_\ell)}(\overrightarrow{x}_1, \ldots, \overrightarrow{x}_h) = 1$, then $m' = m$. Otherwise, $m' \neq m$ except for negligible probability.

For f and f' in $\mathcal{F}$, we denote $f' \leq f$ if the predicate vector for f is a prefix of that for f'. For the following definition for key queries, see [38].

Remark: We will explain the hierarchical structure by using a small (toy) example that has three levels and each level consists of 2-dimensional space, i.e., 6-dimensional space is employed in total. That is, $\overrightarrow{\mu} := (n, d; \mu_1, \ldots, \mu_d) = (6, 3; 2, 4, 6)$ in this example.

A user who possesses a secret key sk_1 in the top level, associated with the top level predicate vector $\overrightarrow{v}_1 := (v_1, v_2)$, can delegate any value (say $\overrightarrow{v}_2 := (v_3, v_4)$) of the second level key sk_2 such that the predicate vector for sk_2 is $(\overrightarrow{v}_1, \overrightarrow{v}_2)$. Similarly, a user who possesses a secret key in the second level, sk_2 with $(\overrightarrow{v}_1, \overrightarrow{v}_2)$, can delegate any value (say $\overrightarrow{v}_3 := (v_5, v_6)$) of the third level key sk_3 with $(\overrightarrow{v}_1, \overrightarrow{v}_2, \overrightarrow{v}_3)$.

Secret key sk_1 with $\overrightarrow{v}_1$, can decrypt a ciphertext associated with attribute vector $(\overrightarrow{x}_1, (*,*), (*,*)) := ((x_1, x_2), (*,*), (*,*))$ if $\overrightarrow{x}_1 \cdot \overrightarrow{v}_1 = 0$, where $*$ denotes an arbitrary value. Secret key sk_2 with $(\overrightarrow{v}_1, \overrightarrow{v}_2)$ can decrypt a ciphertext with attribute vector $(\overrightarrow{x}_1, \overrightarrow{x}_2, (*,*))$ if $\overrightarrow{x}_1 \cdot \overrightarrow{v}_1 = 0$ and $\overrightarrow{x}_2 \cdot \overrightarrow{v}_2 = 0$. However sk_2 cannot decrypt a ciphertext with higher level (top level) attribute vector $\overrightarrow{x}_1 := (x_1, x_2)$ (or $(\overrightarrow{x}_1, (*,*), (*,*))$). Therefore, the capability of a delegated key sk_2 is more limited than the parent key sk_1.

Hence, when $(\overrightarrow{v}_1, \overrightarrow{v}_2) := ((v_1, v_2), (v_3, v_4))$ is a predicate vector for a secret key, $(\overrightarrow{v}_1, \overrightarrow{v}_2)$ is considered to be $(\overrightarrow{v}_1, \overrightarrow{v}_2, (0,0))$, and when $\overrightarrow{x}_1 := (x_1, x_2)$ is an attribute vector for a ciphertext, $\overrightarrow{x}_1$ is considered to be $(\overrightarrow{x}_1, (*,*), (*,*)))$, where $(*,*) \cdot (0,0) = 0$ and $(*,*) \cdot \overrightarrow{v}_2 \neq 0$ unless $\overrightarrow{v}_2 = (0,0)$.

Definition 47 *A hierarchical inner-product predicate encryption scheme for hierarchical predicates* $\mathcal{F}$ *over hierarchical attributes* Σ *is* adaptively attribute-hiding (AH) against chosen plaintext attacks *if for all probabilistic polynomial-time adversaries* $\mathcal{A}$*, the advantage of* $\mathcal{A}$ *in the following experiment is negligible in the security parameter.*

1. Setup *is run to generate keys* pk *and* sk*, and* pk *is given to* $\mathcal{A}$*.*

2. $\mathcal{A}$ *may adaptively makes a polynomial number of queries of the following type:*

   - *[Create key]* $\mathcal{A}$ *asks the challenger to create a secret key for a predicate* $f \in \mathcal{F}$*. The challenger creates a key for* f *without giving it to* $\mathcal{A}$*.*
   - *[Create delegated key]* $\mathcal{A}$ *specifies a key for predicate* f *that has already been created, and asks the challenger to perform a delegation operation to create a child key for* $f' \leq f$*. The challenger computes the child key without giving it to the adversary.*

- *[Reveal key]* $\mathcal{A}$ *asks the challenger to reveal an already-created key for predicate* f.

Note that when key creation requests are made, $\mathcal{A}$ *does not automatically see the created key.* $\mathcal{A}$ *sees a key only when it makes a reveal key query.*

3. $\mathcal{A}$ *outputs challenge attribute vectors* $\mathcal{X}^{(0)} := (\overrightarrow{x}_1^{(0)}, \ldots, \overrightarrow{x}_{h^{(0)}}^{(0)}), \mathcal{X}^{(1)} := (\overrightarrow{x}_1^{(1)}, \ldots, \overrightarrow{x}_{h^{(1)}}^{(1)})$ *and challenge plaintexts* $m^{(0)}, m^{(1)}$*, subject to the restriction that* $f(\mathcal{X}^{(0)}) = f(\mathcal{X}^{(1)}) = 0$ *for all the reveal key queried predicate* f.

4. *A random bit* b *is chosen.* $\mathcal{A}$ *is given* $c^{(b)} \xleftarrow{\mathsf{R}} \mathsf{Enc}(\mathsf{pk}, m^{(b)}, \mathcal{X}^{(b)})$.

5. *The adversary may continue to request keys for additional predicate vectors subject to the restriction that* $f(\mathcal{X}^{(0)}) = f(\mathcal{X}^{(1)}) = 0$.

6. $\mathcal{A}$ *outputs a bit* b'*, and succeeds if* $b' = b$.

We define the advantage of $\mathcal{A}$ *as the quantity* $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{HPE,AH}}(\lambda) := \mathsf{Pr}\left[b' = b\right] - 1/2$.

B.6 Key Idea in Constructing the Proposed HPE

We will explain a key idea of the proposed HPE scheme.

We will explain the key idea of the proposed HPE scheme by using a small (toy) example. Let the dimension of (predicate/attribute) vectors be 6, in which there are three levels and each level has 2-dimensions, $\mathbb{V}$ and $\mathbb{V}^*$ be 15-dimensional spaces, the public parameter be $\widehat{\mathbb{B}} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_6, \boldsymbol{b}_{13}, \boldsymbol{b}_{15})$ as well as the parameters of DPVS, and the master secret key be $\mathbb{B}^* := (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_{15}^*)$.

Ciphertext $(\boldsymbol{c}_1, c_2)$ for attribute $\overrightarrow{x} := (\overrightarrow{x}_1, \overrightarrow{x}_2, \overrightarrow{x}_3) := ((x_1, x_2), (x_3, x_4), (x_5, x_6)) \in \mathbb{F}_q^6$ and plaintext m is constructed as $\boldsymbol{c}_1 := \delta_1(x_1\boldsymbol{b}_1 + x_2\boldsymbol{b}_2) + \cdots + \delta_3(x_5\boldsymbol{b}_5 + x_6\boldsymbol{b}_6) + \zeta\boldsymbol{b}_{13} + \delta_4\boldsymbol{b}_{15}$ and $c_2 := g_T^{\zeta} m$, where $\delta_1, \ldots, \delta_4, \zeta \xleftarrow{\mathsf{U}} \mathbb{F}_q$. If the attribute is a higher level such as $\overrightarrow{x}_1 := (x_1, x_2)$, generate a modified attribute $\overrightarrow{x}^+ := ((x_1, x_2), (x_3^+, x_4^+), (x_5^+, x_6^+))$, where $(x_3^+, x_4^+, x_5^+, x_6^+) \xleftarrow{\mathsf{U}} \mathbb{F}_q^4$. Then, ciphertext $\boldsymbol{c}_1$ for attribute $\overrightarrow{x}_1$ is computed as ciphertext $\boldsymbol{c}_1$ for the modified attribute $\overrightarrow{x}^+$.

Top level secret key $\overrightarrow{\boldsymbol{k}}_1^* := (\boldsymbol{k}_{1,\mathsf{dec}}^*, \boldsymbol{k}_{1,\mathsf{ran},1}^*, \boldsymbol{k}_{1,\mathsf{ran},2}^*, \boldsymbol{k}_{1,\mathsf{del},3}^*, \ldots, \boldsymbol{k}_{1,\mathsf{del},6}^*)$, for predicate $\overrightarrow{v} := (v_1, v_2) \in \mathbb{F}_q^2$ consists of three parts, $\boldsymbol{k}_{1,\mathsf{dec}}^*$, $(\boldsymbol{k}_{1,\mathsf{ran},1}^*, \boldsymbol{k}_{1,\mathsf{ran},2}^*)$ and $(\boldsymbol{k}_{1,\mathsf{del},3}^*, \ldots, \boldsymbol{k}_{1,\mathsf{del},6}^*)$, where the first one is used for decryption of ciphertexts, the second one for re-randomization (of delegated key) and the last one for delegation. Each part is: $\boldsymbol{k}_{1,\mathsf{dec}}^* := \sigma_{1,\mathsf{dec}}(v_1\boldsymbol{b}_1^* + v_2\boldsymbol{b}_2^*) + \boldsymbol{b}_{13}^* + \eta_{1,\mathsf{dec}}\boldsymbol{b}_{14}^*$, $\boldsymbol{k}_{1,\mathsf{ran},j}^* := \sigma_{1,\mathsf{ran},j}(v_1\boldsymbol{b}_1^* + v_2\boldsymbol{b}_2^*) + \eta_{1,\mathsf{ran},j}\boldsymbol{b}_{14}^*$ $(j = 1, 2)$, and $\boldsymbol{k}_{1,\mathsf{del},j}^* := \sigma_{1,\mathsf{del},j}(v_1\boldsymbol{b}_1^* + v_2\boldsymbol{b}_2^*) + \psi\boldsymbol{b}_j + \eta_{1,\mathsf{del},j}\boldsymbol{b}_{14}^*$ $(j = 3, \ldots, 6)$, where the coefficients are uniformly selected from $\mathbb{F}_q$. The first one, $\boldsymbol{k}_{1,\mathsf{dec}}^*$, can decrypt ciphertext $(\boldsymbol{c}_1, c_2)$ by $c_2/e(\boldsymbol{c}_1, \boldsymbol{k}_{1,\mathsf{dec}}^*)$, since $e(\boldsymbol{c}_1, \boldsymbol{k}_{1,\mathsf{dec}}^*) = g_T^{\zeta}$ if an attribute of $\boldsymbol{c}_1$ is $((x_1, x_2), (*, *), (*, *))$ with $(x_1, x_2) \cdot (v_1, v_2) = 0$.

To delegate a secret key for the 2nd level vector (v_3, v_4), $\sigma_{\mathsf{dec}}(v_3\boldsymbol{k}_{1,\mathsf{del},3}^* + v_4\boldsymbol{k}_{1,\mathsf{del},4}^*)$ is added to $\boldsymbol{k}_{1,\mathsf{dec}}^*$ $(j = 0)$, and $\sigma_{\mathsf{del},j}(v_3\boldsymbol{k}_{1,\mathsf{del},3}^* + v_4\boldsymbol{k}_{1,\mathsf{del},4}^*)$ is added to $\boldsymbol{k}_{1,\mathsf{del},j}^*$ $(j = 5, 6)$. To re-randomize the coefficients of $(v_1\boldsymbol{b}_1^* + v_2\boldsymbol{b}_2^*)$, and $\boldsymbol{b}_{14}^*$ in the delegated key, a random linear combination of $\boldsymbol{k}_{1,\mathsf{ran},1}^*$ and $\boldsymbol{k}_{1,\mathsf{ran},2}^*$ is also added. So, the delegated key (the second level key) is $\overrightarrow{\boldsymbol{k}}_2^* := (\boldsymbol{k}_{2,\mathsf{dec}}^*, \boldsymbol{k}_{2,\mathsf{ran},1}^*, \ldots, \boldsymbol{k}_{2,\mathsf{ran},3}^*, \boldsymbol{k}_{2,\mathsf{del},5}^*, \boldsymbol{k}_{2,\mathsf{del},6}^*)$, (where $\boldsymbol{k}_{2,\mathsf{dec}}^*$ is for decryption, $(\boldsymbol{k}_{2,\mathsf{ran},1}^*, \ldots, \boldsymbol{k}_{2,\mathsf{ran},3}^*)$ for re-randomization, and $(\boldsymbol{k}_{2,\mathsf{del},5}^*, \boldsymbol{k}_{2,\mathsf{del},6}^*)$ for delegation) is computed as $\boldsymbol{k}_{2,\mathsf{dec}}^* := \boldsymbol{k}_{1,\mathsf{dec}}^* + (\alpha_{\mathsf{dec},1}\boldsymbol{k}_{1,\mathsf{ran},1}^* + \alpha_{\mathsf{dec},2}\boldsymbol{k}_{1,\mathsf{ran},2}^*) + \sigma_{\mathsf{dec}}(v_3\boldsymbol{k}_{1,\mathsf{del},3}^* + v_4\boldsymbol{k}_{1,\mathsf{del},4}^*)$, $\boldsymbol{k}_{2,\mathsf{ran},j}^* := (\alpha_{\mathsf{ran},j,1}\boldsymbol{k}_{1,\mathsf{ran},1}^* + \alpha_{\mathsf{ran},j,2}\boldsymbol{k}_{1,\mathsf{ran},2}^*) + \sigma_{\mathsf{ran},j}(v_3\boldsymbol{k}_{1,\mathsf{del},3}^* + v_4\boldsymbol{k}_{1,\mathsf{del},4}^*)$ $(j = 1, 2, 3)$, and $\boldsymbol{k}_{2,\mathsf{del},j}^* := \psi^+\boldsymbol{k}_{1,\mathsf{del},j}^* + (\alpha_{\mathsf{del},j,1}\boldsymbol{k}_{1,\mathsf{ran},1}^* + \alpha_{\mathsf{del},j,2}\boldsymbol{k}_{1,\mathsf{ran},2}^*) + \sigma_{\mathsf{del},j}(v_3\boldsymbol{k}_{1,\mathsf{del},3}^* + v_4\boldsymbol{k}_{1,\mathsf{del},4}^*)$ $(j = 5, 6)$, where the coefficients are uniformly selected from $\mathbb{F}_q$. Then, the distribution of the delegated key (by $\mathsf{Delegate}$) is equivalent to that obtained by

the key generation query (KeyGen) except negligible probability (i.e., the simulation of 'create delegated key query' can be equivalent to that of 'create key query'.)

[Correctness of the Proposed HPE Scheme] Assume that ciphertext $(\boldsymbol{c}_1, c_2)$ is generated by $\mathsf{Enc}(\mathsf{pk}, m, (\overrightarrow{x}_1, \ldots, \overrightarrow{x}_h))$ and secret key $\boldsymbol{k}^*_{\ell,\mathsf{dec}}$ is generated by $\mathsf{KeyGen}(\mathsf{pk}, \mathsf{sk}, (\overrightarrow{v}_1, \ldots, \overrightarrow{v}_\ell))$. Note that $e(\boldsymbol{c}_1, \boldsymbol{k}^*_{\ell,\mathsf{dec}}) = g_T^{\sum_{1\le i\le \ell} \sigma_i \delta_i \overrightarrow{x}_i \cdot \overrightarrow{v}_i + \zeta}$. If $\ell \le h$ and $\overrightarrow{x}_i \cdot \overrightarrow{v}_i = 0$ for all i s.t. $1 \le i \le \ell$, then $e(\boldsymbol{c}_1, \boldsymbol{k}^*_{\ell,\mathsf{dec}}) = g_T^{\zeta}$. Otherwise, $e(\boldsymbol{c}_1, \boldsymbol{k}^*_{\ell,\mathsf{dec}})$ is uniformly distributed. Hence, correctness holds for secret keys generated by KeyGen, and it also holds for keys generated by $\mathsf{Delegate}$ by Claim 3.

B.7 Proof of Theorem 28

Proof Outline: To prove the security, we employ Game 0 (original adaptive-security game) through Game 4. Roughly speaking, a delegated key query (i.e., a reveal query of an already-created delegated key) is replied by using KeyGen (in place of $\mathsf{Delegate}$) in Game 1, the (normal) target ciphertext is changed to a *semi-functional* ciphertext in Game 2, the j-th key (normal key) in the k-th reveal key query's reply (including $\ell + n - \mu_\ell + 2 \le n + 1$ keys, where ℓ is the level of the k-th queried key) is changed to a *semi-functional* key in Game 3-(k, j) ($k = 1, \ldots, \nu;\ j = 1, \ldots, n+1$), and the (semi-functional) target ciphertext is changed to perfectly *randomized* key in Game 4, whose advantage is 0.

Since the distribution regarding each revealed key query in Game 1 is equivalent to that in Game 0 except with probability at most $3/q$, the gap between Games 0 and 1 is bounded by $3\nu/q$.

To prove that the gap between Games 1 and 2 is bounded by the advantage of Problem 1 (to guess $\beta \in \{0, 1\}$), we show that the key query replies and target ciphertext in Game 1 or 2 are generated by using the Problem 1 instance such that the distribution is equivalent to those of Game 1 when $\beta = 0$ and is equivalent to Game 2 when $\beta = 1$. The advantage of Problem 1 is shown to be equivalent to the advantage of our assumption, the n-eDDH assumption.

The gap between Games 3-$(k, j-1)$ and 3-(k, j) is similarly shown to be bounded by the advantage of Problem 2 (i.e., of the n-eDDH assumption).

Finally we show that Game 3-$(\nu, n+1)$ can be conceptually changed to Game 4 by using the fact that n elements of $\mathbb{B}$, $(\boldsymbol{b}_{n+1}, \ldots, \boldsymbol{b}_{2n})$, are secret to the adversary.

Proof of Theorem 28: To prove Theorem 28, we consider the following games.

Game 0: Original game (Definition 47).

Game 1: Game 1 is the same as Game 0 except the following procedures.

1. When a create key query is issued by $\mathcal{A}$, the challenger of the game only records the specified predicates, and when a create delegated key query is issued, the challenger only records the specified keys and predicates. In this step, just the query is recorded, but no corresponding key is created.
2. When a reveal key query is issued for a hierarchical (level-ℓ) predicate $(\overrightarrow{v}_1, \ldots, \overrightarrow{v}_\ell)$ which has been already recorded, the challenger creates the queried key by using KeyGen.

Game 2: Game 2 is the same as Game 1 except that the target ciphertext is $(\boldsymbol{c}_1^{\mathsf{semi}}, c_2)$:

$$\boldsymbol{c}_1^{\mathsf{semi}} := \boldsymbol{c}_1^{\mathsf{norm}} + (w_1 \boldsymbol{b}_{n+1} + \cdots + w_n \boldsymbol{b}_{2n}),$$

where $(\boldsymbol{c}_1^{\mathsf{norm}}, c_2)$ is a correctly generated target ciphertext for adversary $\mathcal{A}$'s encryption query, and $(w_1, \ldots, w_n) \xleftarrow{\mathsf{U}} \mathbb{F}_q^n \setminus \{\overrightarrow{0}\}$.

Game 3-(k,j) ($k = 1, \ldots, \nu$; $j = 1, \ldots, n+1$): Game 3-$(1,0)$ is Game 2. The number of keys, $(\boldsymbol{k}^*_{\ell,\mathsf{dec}}, \boldsymbol{k}^*_{\ell,\mathsf{ran},1}, \ldots, \boldsymbol{k}^*_{\ell,\mathsf{ran},\ell+1}, \boldsymbol{k}^*_{\ell,\mathsf{del},\mu_\ell+1}, \ldots, \boldsymbol{k}^*_{\ell,\mathsf{del},n})$, which is a reply to the k-th reveal key query in the game, is less than or equal to $n+1$, i.e., $\ell + n - \mu_\ell + 2 \leq n+1$. Game 3-$(k, n+1)$ is Game 3-$(k+1, 0)$.

Game 3-(k,j) is the same as Game 3-$(k, j-1)$ except that the j-th key in the k-th reveal key query's reply is $\boldsymbol{k}^{*\mathsf{semi}}_{(k,j)}$:

$$\boldsymbol{k}^{*\mathsf{semi}}_{(k,j)} := \boldsymbol{k}^{*\mathsf{norm}}_{(k,j)} + (r_1 \boldsymbol{b}^*_{n+1} + \cdots + r_n \boldsymbol{b}^*_{2n}),$$

where $\boldsymbol{k}^{*\mathsf{norm}}_{(k,j)}$ is a correctly generated value of the j-th key in a reply to the k-th reveal key query (i.e., $\boldsymbol{k}^{*\mathsf{norm}}_{(k,j)}$ is the j-th key in a reply to the k-th reveal key query in Game 3-$(k, j-1)$), and $(r_1, \ldots, r_n) \xleftarrow{\mathsf{U}} \mathbb{F}_q^n$.

Game 4 Game 4 is the same as Game 3-$(\nu, n+1)$ except that the target ciphertext is $(\boldsymbol{c}_1^{\mathsf{rand}}, c_2)$:

$$\boldsymbol{c}_1^{\mathsf{rand}} := \boldsymbol{c}_1^{\mathsf{semi}} + (x'_1 \boldsymbol{b}_1 + \cdots + x'_n \boldsymbol{b}_n) + \zeta' \boldsymbol{b}_{2n+1},$$

where $x'_1, \ldots, x'_n, \zeta' \xleftarrow{\mathsf{U}} \mathbb{F}_q$.

Let $\mathsf{Adv}^{(0)}_{\mathcal{A}}(\lambda)$, $\mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda)$, $\mathsf{Adv}^{(2)}_{\mathcal{A}}(\lambda)$, $\mathsf{Adv}^{(3\text{-}(k,j))}_{\mathcal{A}}(\lambda)$ and $\mathsf{Adv}^{(4)}_{\mathcal{A}}(\lambda)$ be $\mathsf{Adv}^{\mathsf{HPE,AH}}_{\mathcal{A}}(\lambda)$ in Game 0, Game 1, Game 2, Game 3-(k,j) and Game 4 ($k = 1, \ldots, \nu; j = 1, \ldots, n+1$). It is clear that $\mathsf{Adv}^{(4)}_{\mathcal{A}}(\lambda) = 0$.

We will use lemmas (Lemmas 48, 49, 50, 51) that evaluate the gaps between pairs of the neighboring advantages. From these lemmas as well as Lemmas 20 and 22, we obtain

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{HPE,AH}}_{\mathcal{A}}(\lambda) &\leq |\mathsf{Adv}^{(0)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda)| + |\mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(2)}_{\mathcal{A}}(\lambda)| \\
&\quad + \sum_{k=1}^{\nu} \sum_{j=1}^{n+1} |\mathsf{Adv}^{(3\text{-}(k,j-1))}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(3\text{-}(k,j))}_{\mathcal{A}}(\lambda)| + |\mathsf{Adv}^{(3\text{-}(\nu,n+1))}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(4)}_{\mathcal{A}}(\lambda)| \\
&< \frac{3\nu}{q} + \mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}_0}(\lambda) + \sum_{k=1}^{\nu} \sum_{j=1}^{n+1} (\mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}_{(k,j)}}(\lambda) + \frac{1}{q}) \\
&\leq \mathsf{Adv}^{n\text{-}\mathsf{eDDH}}_{\mathcal{C}_0}(\lambda) + \sum_{k=1}^{\nu} \sum_{j=1}^{n+1} \mathsf{Adv}^{n\text{-}\mathsf{eDDH}}_{\mathcal{C}_{(k,j)}}(\lambda) + \frac{(n+4)\nu}{q}.
\end{aligned}$$

This completes the proof of Theorem 28. □

Lemma 48 *For any adversary $\mathcal{A}$, $|\mathsf{Adv}^{(0)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda)| < 3\nu/q$.*

Proof. The distribution of $\overrightarrow{\boldsymbol{k}}^*_{\ell+1}$ generated by KeyGen for a level-$(\ell+1)$ predicate is equivalent to that by the combination of KeyGen for the level-ℓ predicate and $\mathsf{Delegate}_\ell$ except with probability $3/q$, from Claim 3. Therefore, the revealed key distribution in Game 0 is equivalent to that in Game 1 except with probability at most $(1 - (1 - 3/q)^\nu) \leq 3\nu/q$, since the number of delegate queries is upper-bounded by ν. Hence (by using Shoup's difference lemma), the difference of $\mathsf{Adv}^{(0)}_{\mathcal{A}}(\lambda)$ and $\mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda)$ is upper-bounded by $3\nu/q$. □

Claim 3 *If* $\overrightarrow{\boldsymbol{k}}^*_\ell$ *is generated by* $\mathsf{KeyGen}(\mathsf{pk},\mathsf{sk},(\overrightarrow{v}_1,\ldots,\overrightarrow{v}_\ell))$*, the distribution of* $\overrightarrow{\boldsymbol{k}}^*_{\ell+1}$ *generated by* $\mathsf{Delegate}(\mathsf{pk},\overrightarrow{\boldsymbol{k}}^*_\ell,\overrightarrow{v}_{\ell+1})$ *is equivalent to that of* $\overrightarrow{\boldsymbol{k}}^*_{\ell+1}$ *generated by* $\mathsf{KeyGen}(\mathsf{pk},\mathsf{sk},(\overrightarrow{v}_1,..,\overrightarrow{v}_\ell,\overrightarrow{v}_{\ell+1}))$ *except with probability at most* $3/q$*.*

Proof. The distribution of level-ℓ key $\boldsymbol{k}^*_{\ell,\mathsf{ran},j}$ $(j=1,\ldots,\ell+1)$ is represented by that of the $\ell+1$ coefficients, $(\sigma_{\mathsf{ran},j,1},\ldots,\sigma_{\mathsf{ran},j,\ell},\eta_{\mathsf{ran},j})$, of $\sum_{i=\mu_{t-1}+1}^{\mu_t} v_i\boldsymbol{b}^*_{\ell,i}$ $(t=1,\ldots,\ell)$ and $\boldsymbol{b}^*_{n+1}$ (and the coefficient, ψ, of $\boldsymbol{b}^*_j$ in addition for $\boldsymbol{k}^*_{\ell,\mathsf{del},j}$ when $j=\mu_\ell+1,\ldots,n$).

Similarly, the distribution of level-$(\ell+1)$ key $\boldsymbol{k}^*_{\ell+1,\mathsf{ran},j}$ $(j=1,\ldots,\ell+2)$ is represented by that of the $\ell+2$ coefficients, $(\sigma_{\mathsf{ran},j,1},\ldots,\sigma_{\mathsf{ran},j,\ell+1},\eta_{\mathsf{ran},j})$.

When level-ℓ key $\boldsymbol{k}^*_{\ell,\mathsf{ran},j}$ $(j=1,\ldots,\ell+1)$ is generated by $\mathsf{KeyGen}(\mathsf{pk},\mathsf{sk},(\overrightarrow{v}_1,\ldots,\overrightarrow{v}_\ell))$, coefficients $(\sigma_{\mathsf{ran},j,1},\ldots,\sigma_{\mathsf{ran},j,\ell},\eta_{\mathsf{ran},j})_{j=1,\ldots,\ell+1}$ are uniformly distributed.

If coefficient matrix $(\sigma_{\mathsf{ran},j,1},\ldots,\sigma_{\mathsf{ran},j,\ell},\eta_{\mathsf{ran},j})_{j=1,\ldots,\ell+1}$ ($(\ell+1)\times(\ell+1)$ matrix) of $(\boldsymbol{k}^*_{\ell,\mathsf{ran},j})_{j=1,..,\ell+1}$ is regular and $\psi\neq 0$, then the coefficients, $(\sigma_{\mathsf{ran},j,1},\ldots,\sigma_{\mathsf{ran},j,\ell+1},\eta_{\mathsf{ran},j})$, of $\mathsf{Delegate}(\mathsf{pk},\overrightarrow{\boldsymbol{k}}^*_\ell,\overrightarrow{v}_{\ell+1})$ are uniformly distributed, i.e., $\mathsf{Delegate}(\mathsf{pk},\overrightarrow{\boldsymbol{k}}^*_\ell,\overrightarrow{v}_{\ell+1})$ is equivalently distributed as $\mathsf{KeyGen}(\mathsf{pk},\mathsf{sk},(\overrightarrow{v}_1,\ldots,\overrightarrow{v}_{\ell+1}))$.

Here, $(\sigma_{\mathsf{ran},j,1},\ldots,\sigma_{\mathsf{ran},j,\ell},\eta_{\mathsf{ran},j})_{j=1,\ldots,\ell+1}$ ($(\ell+1)\times(\ell+1)$ matrix) of $(\boldsymbol{k}^*_{\ell,\mathsf{ran},j})_{j=1,\ldots,\ell+1}$ is regular and $\psi\neq 0$ except with probability at most $2/q+1/q=3/q$, from the following claim (since obviously $q>2$). □

Claim 4 *Let* $q>2$ *and* $\Delta:=\{M|\det M\neq 0\}\subset\mathbb{F}_q^{s\times s}$*. Then,*

$$\frac{|\Delta|}{q^{s^2}}<\frac{2}{q}.$$

Proof. Since $|\Delta|=q^{s^2}-|GL(s,\mathbb{F}_q)|$, we will show inequality (2), i.e., a lower bound of $|GL(s,\mathbb{F}_q)|$.

$$|GL(s,\mathbb{F}_q)|>q^{s^2}-\frac{q^{s^2}}{q-1}. \tag{2}$$

From the following well-known formula,

$$\begin{aligned}|GL(s,\mathbb{F}_q)| &= (q^s-1)(q^s-q)\cdots(q^s-q^{s-1})\\ &= q^{s(s-1)/2}(q^s-1)(q^{s-1}-1)\cdots(q-1)\\ &= q^{s^2}(1-q^{-1})(1-q^{-2})\cdots(1-q^{-s}),\end{aligned}$$

we obtain

$$\begin{aligned}|GL(s,\mathbb{F}_q)| &= q^{s^2}(1-q^{-1})(1-q^{-2})\cdots(1-q^{-s})\\ &= q^{s^2}\left(1+\sum_{k=1}^{s}(-1)^k\sum_{1\le i_1<\cdots<i_k\le s}q^{-(i_1+\cdots+i_k)}\right)\\ &= q^{s^2}-q^{s^2}\sum_{1\le i\le s}q^{-i}+q^{s^2}\sum_{k=2}^{s}(-1)^k\sum_{1\le i_1<\cdots<i_k\le s}q^{-(i_1+\cdots+i_k)}\\ &= q^{s^2}-q^{s^2}\cdot\frac{1}{q^s}\frac{q^s-1}{q-1}+q^{s^2}\sum_{k=2}^{s}(-1)^k\sum_{1\le i_1<\cdots<i_k\le s}q^{-(i_1+\cdots+i_k)}\\ &> q^{s^2}-\frac{q^{s^2}}{q-1}+q^{s^2}\sum_{k=2}^{s}(-1)^k\sum_{1\le i_1<\cdots<i_k\le s}q^{-(i_1+\cdots+i_k)}\end{aligned}$$

Let $u := \lfloor s/2 \rfloor$ when s is odd, and $u := s/2 - 1$ when s is even. Then, the above expression is

$$\begin{aligned}
&> q^{s^2} - \frac{q^{s^2}}{q-1} + q^{s^2} \sum_{t=1}^{u} \left(\sum_{1 \le i_1 < \cdots < i_{2t} \le s} q^{-(i_1 + \cdots + i_{2t})} - \sum_{1 \le i_1 < \cdots < i_{2t+1} \le s} q^{-(i_1 + \cdots + i_{2t+1})} \right) \\
&= q^{s^2} - \frac{q^{s^2}}{q-1} + q^{s^2} \sum_{t=1}^{u} \sum_{1 \le i_1 < \cdots < i_{2t} \le s} q^{-(i_1 + \cdots + i_{2t})} \left(1 - \sum_{i_{2t} < i_{2t+1} \le s} q^{-i_{2t+1}} \right) \\
&> q^{s^2} - \frac{q^{s^2}}{q-1} + q^{s^2} \sum_{t=1}^{u} \sum_{i_1 < \cdots < i_{2t}} q^{-(i_1 + \cdots + i_{2t})} \left(1 - \sum_{1 \le i \le s} q^{-i} \right) \\
&= q^{s^2} - \frac{q^{s^2}}{q-1} + q^{s^2} \sum_{t=1}^{u} \sum_{i_1 < \cdots < i_{2t}} q^{-(i_1 + \cdots + i_{2t})} \left(1 - \frac{q^s - 1}{q^s (q-1)} \right) \\
&> q^{s^2} - \frac{q^{s^2}}{q-1}.
\end{aligned}$$

This shows (2). Hence,

$$|\Delta| \;=\; q^{s^2} - |GL(s, \mathbb{F}_q)| < \frac{q^{s^2}}{q-1}.$$

Therefore,

$$\frac{|\Delta|}{q^{s^2}} < \frac{1}{q-1} < \frac{2}{q} \quad \text{since} \quad q > 2.$$

This completes the proof of Claim 4. □

Lemma 49 *For any adversary $\mathcal{A}$, there exists a probabilistic machine $\mathcal{B}_1$, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $|\mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(2)}_{\mathcal{A}}(\lambda)| = \mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}_0}(\lambda)$.*

Since the proof of this lemma is essentially the same as that for Lemma 24, we omit the proof.

Lemma 50 *For any adversary $\mathcal{A}$, there exists a probabilistic machine $\mathcal{B}$, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $|\mathsf{Adv}^{(3\text{-}(k,j-1))}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(3\text{-}(k,j))}_{\mathcal{A}}(\lambda)| \le \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}_{(k,j)}}(\lambda) + \frac{1}{q}$.*

Since the proof of this lemma is essentially the same as that for Lemma 25, we omit the proof.

Lemma 51 *For any adversary $\mathcal{A}$, $\mathsf{Adv}^{(3\text{-}(\nu,n+1))}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{(4)}_{\mathcal{A}}(\lambda)$.*

Since the proof of this lemma is essentially the same as that for Lemma 26, we omit the proof.

B.8 General Delegation

A generalized delegation (not limited to a hierarchical delegation) system can be constructed in a manner similar to the hierarchical delegation.

$\mathsf{Setup}(1^\lambda, \overrightarrow{\mu} := (n, d; \mu_1, \ldots, \mu_d)) :\ (\mathsf{param}_{\mathbb{V}}, \mathbb{B}, \mathbb{B}^*) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^\lambda, 2n+3)$,
$\widehat{\mathbb{B}} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n, \boldsymbol{b}_{2n+1}, \boldsymbol{b}_{2n+3})$, $\mathsf{sk} := \mathbb{B}^*$, $\mathsf{pk} := (1^\lambda, \mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}})$.
return sk, pk.

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{sk}, (\overrightarrow{v}_1, \ldots, \overrightarrow{v}_\ell) := ((v_{1,1}, \ldots, v_{1,n}), \ldots, (v_{\ell,1}, \ldots, v_{\ell,n})) :$
$\sigma_{\mathsf{dec},t}, \eta_{\mathsf{dec}}$, $\sigma_{\mathsf{ran},j,t}, \eta_{\mathsf{ran},j}$ $(j = 1, \ldots, \ell+1)$, $\sigma_{\mathsf{del},j,t}, \eta_{\mathsf{del},j}$ $(j = 1, \ldots, n)$, $\psi \xleftarrow{\mathsf{U}} \mathbb{F}_q$
for $t = 1, \ldots, \ell$,
$\boldsymbol{k}^*_{\ell,\mathsf{dec}} := \sum_{t=1}^{\ell} \sigma_{\mathsf{dec},t}(\sum_{i=1}^{n} v_{t,i}\boldsymbol{b}^*_i) + \boldsymbol{b}^*_{2n+1} + \eta_{\mathsf{dec}}\boldsymbol{b}^*_{2n+2}$,
$\boldsymbol{k}^*_{\ell,\mathsf{ran},j} := \sum_{t=1}^{\ell} \sigma_{\mathsf{ran},j,t}(\sum_{i=1}^{n} v_{t,i}\boldsymbol{b}^*_i) + \eta_{\mathsf{ran},j}\boldsymbol{b}^*_{2n+2}$
for $j = 1, \ldots, \ell+1$,
$\boldsymbol{k}^*_{\ell,\mathsf{del},j} := \sum_{t=1}^{\ell} \sigma_{\mathsf{del},j,t}(\sum_{i=1}^{n} v_{t,i}\boldsymbol{b}^*_i) + \psi\boldsymbol{b}^*_j + \eta_{\mathsf{del},j}\boldsymbol{b}^*_{2n+2}$
for $j = 1, \ldots, n$,
return $\overrightarrow{\boldsymbol{k}}^*_\ell := (\boldsymbol{k}^*_{\ell,\mathsf{dec}}, \boldsymbol{k}^*_{\ell,\mathsf{ran},1}, \ldots, \boldsymbol{k}^*_{\ell,\mathsf{ran},\ell+1}, \boldsymbol{k}^*_{\ell,\mathsf{del},1}, \ldots, \boldsymbol{k}^*_{\ell,\mathsf{del},n})$.

$\mathsf{Enc}(\mathsf{pk}, m \in \mathbb{G}_T, (\overrightarrow{x}_1, \ldots, \overrightarrow{x}_\ell) := ((x_{1,1}, \ldots, x_{1,n}), \ldots, (x_{\ell,1}, \ldots, x_{\ell,n})) :$
$\delta_1, \ldots, \delta_\ell, \delta_{2n+3}, \zeta \xleftarrow{\mathsf{U}} \mathbb{F}_q$,
$\boldsymbol{c}_1 := \sum_{t=1}^{\ell} \delta_t(\sum_{i=1}^{n} x_{t,i}\boldsymbol{b}_i) + \zeta\boldsymbol{b}_{2n+1} + \delta_{2n+3}\boldsymbol{b}_{2n+3}$, $c_2 := g_T^{\zeta} m$,
return $(\boldsymbol{c}_1, c_2)$.

$\mathsf{Dec}(\mathsf{pk}, \boldsymbol{k}^*_{\ell,\mathsf{dec}}, \boldsymbol{c}_1, c_2) :\ m' := c_2 / e(\boldsymbol{c}_1, \boldsymbol{k}^*_{\ell,\mathsf{dec}})$,
return m'.

$\mathsf{Delegate}_\ell(\mathsf{pk}, \overrightarrow{\boldsymbol{k}}^*_\ell, \overrightarrow{v}_{\ell+1} := (v_{\ell+1,1}, \ldots, v_{\ell+1,n}) \notin \mathsf{span}\langle \overrightarrow{v}_1, \ldots, \overrightarrow{v}_\ell \rangle) :$
$\alpha_{\mathsf{dec},t}, \sigma_{\mathsf{dec}}$, $\alpha_{\mathsf{ran},j,t}, \sigma_{\mathsf{ran},j}$ $(j = 1, \ldots, \ell+2)$, $\alpha_{\mathsf{del},j,t}, \sigma_{\mathsf{del},j}$ $(j = 1, \ldots, n)$, $\psi' \xleftarrow{\mathsf{U}} \mathbb{F}_q$
for $t = 1, \ldots, \ell+1$,
$\boldsymbol{k}^*_{\ell+1,\mathsf{dec}} := \boldsymbol{k}^*_{\ell,\mathsf{dec}} + \sum_{t=1}^{\ell+1} \alpha_{\mathsf{dec},t}\boldsymbol{k}^*_{\ell,\mathsf{ran},t} + \sigma_{\mathsf{dec}}(\sum_{i=1}^{n} v_{\ell+1,i}\boldsymbol{k}^*_{\ell,\mathsf{del},i})$,
$\boldsymbol{k}^*_{\ell+1,\mathsf{ran},j} := \sum_{t=1}^{\ell+1} \alpha_{\mathsf{ran},j,t}\boldsymbol{k}^*_{\ell,\mathsf{ran},t} + \sigma_{\mathsf{ran},j}(\sum_{i=1}^{n} v_{\ell+1,i}\boldsymbol{k}^*_{\ell,\mathsf{del},i})$ for $j = 1, \ldots, \ell+2$,
$\boldsymbol{k}^*_{\ell+1,\mathsf{del},j} := \sum_{t=1}^{\ell+1} \alpha_{\mathsf{del},j,t}\boldsymbol{k}^*_{\ell,\mathsf{ran},t} + \sigma_{\mathsf{del},j}(\sum_{i=1}^{n} v_{\ell+1,i}\boldsymbol{k}^*_{\ell,\mathsf{del},i}) + \psi'\boldsymbol{k}^*_{\ell,\mathsf{del},j}$ for $j = 1, \ldots, n$,
return $\overrightarrow{\boldsymbol{k}}^*_{\ell+1} := (\boldsymbol{k}^*_{\ell+1,\mathsf{dec}}, \boldsymbol{k}^*_{\ell+1,\mathsf{ran},1}, \ldots, \boldsymbol{k}^*_{\ell+1,\mathsf{ran},\ell+2}, \boldsymbol{k}^*_{\ell+1,\mathsf{del},1}, \ldots, \boldsymbol{k}^*_{\ell+1,\mathsf{del},n})$.

Let $(\boldsymbol{c}_1, c_2)$ be a ciphertext for attribute $(\overrightarrow{x}_1, \ldots, \overrightarrow{x}_k)$ (and plaintext $m \in \mathbb{G}_T$) and $\overrightarrow{\boldsymbol{k}}^*_\ell$ be a key for predicate $(\overrightarrow{v}_1, \ldots, \overrightarrow{v}_\ell)$. Then, $\overrightarrow{\boldsymbol{k}}^*_\ell$ decrypts $(\boldsymbol{c}_1, c_2)$ if $\overrightarrow{v}_i \cdot \overrightarrow{x}_j = 0$ for all $1 \le i \le \ell$ and $1 \le j \le k$. Namely the capability of delegated key $\overrightarrow{\boldsymbol{k}}^*_{\ell+1}$ is more limited than that of its parent key $\overrightarrow{\boldsymbol{k}}^*_\ell$.