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

Allison Lewko[1,*], Tatsuaki Okamoto[2], Amit Sahai[3,**],
Katsuyuki Takashima[4], and Brent Waters[5,***]

[1] University of Texas at Austin
alewko@cs.utexas.edu
[2] NTT
okamoto.tatsuaki@lab.ntt.co.jp
[3] UCLA
sahai@cs.ucla.edu
[4] Mitsubishi Electric
Takashima.Katsuyuki@aj.MitsubishiElectric.co.jp
[5] University of Texas at Austin
bwaters@cs.utexas.edu

Abstract. We present two fully secure functional encryption schemes: a fully secure attribute-based encryption (ABE) scheme and a fully secure (attribute-hiding) predicate encryption (PE) scheme for inner-product predicates. In both cases, previous constructions were only proven to be selectively secure. Both results use novel strategies to adapt the dual system encryption methodology introduced by Waters. We construct our ABE scheme in composite order bilinear groups, and prove its security from three static assumptions. Our ABE scheme supports arbitrary monotone access formulas. Our predicate encryption scheme is constructed via a new approach on bilinear pairings using the notion of dual pairing vector spaces proposed by Okamoto and Takashima.

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

[*] 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 PRESIDIO), Department of Homeland Security Grant 2006-CS-001-000001-02 (subaward 641), and the Alfred P. Sloan Foundation.

H. Gilbert (Ed.): EUROCRYPT 2010, LNCS 6110, pp. 62–91, 2010.

© International Association for Cryptologic Research 2010

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. [34,25,17,5,32,24,39,27,12]. 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 [34] 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 [27]. 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 [40] 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 [34,25,18,5,32,24,39] 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 [38] 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 [40,28] to the ABE case. Waters [40] 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 [28] 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 [40,28] 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 [28], 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 [28].

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 [27] as a generalized (fine-grained) notion of encryption that covers identity-based encryption (IBE) [6,7,9,19,21,26], hidden-vector encryption (HVE) [12] and attribute-based encryption (ABE) [5,25,32,33,34]. 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 [27,31].
- 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 [16] or that by Boneh and Katz [11] (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 [40] and a new approach based on a notion of higher dimensional vector spaces, *dual pairing vector spaces* (DPVS), proposed by Okamoto and Takashima [30,31]. The notion of DPVS is constructed on bilinear pairing groups, and they presented a selectively secure (H)PE scheme on DPVS [31]. 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 [31] 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 [13], 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 [35]. 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 [19]. These schemes were proven secure in the random oracle model. Then selectively secure schemes in the standard model were constructed [15,6]. Boneh and Boyen [7] and Waters [38] constructed fully secure IBE schemes in the standard model. Gentry [21] 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) [23,26] 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 [22] extended Gentry’s techniques to get a fully secure HIBE system, but under “q-type” assumptions. Waters [40] leveraged the dual system encryption methodology to obtain fully secure IBE and HIBE systems from simple assumptions. Lewko and Waters [28] 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 [34]. Goyal, Pandey, Sahai, and Waters [25] 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 [34,25,18,5,32,24,39].

Goyal, Jain, Pandey, and Sahai [24] provide a general way to transform a KP-ABE system into a CP-ABE system. Chase [17] considered the problem of ABE with multiple authorities.

Other works have discussed similar problems without addressing collusion resistance [1,2,3,14,29,37]. 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 [27], 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 [36] 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 [27]. Moreover, they proved only selective security.

Okamoto and Takashima [31] 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 [30,31], 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 linear secret-sharing schemes (LSSS), CP-ABE, and 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

Linear Secret-Sharing Schemes. The formal definitions of access structures and linear secret-sharing schemes (LSSS) can be found in [4] and the full version of this paper. Informally, a LSSS is a share-generating matrix A whose rows are labeled by attributes. 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. A user's set of attributes S satisfies the LSSS access matrix if the rows labeled by the attributes in S have the *linear reconstruction* property, which means there exist constants $\{\omega_i\}$ such that, for any valid shares $\{\lambda_i\}$ of a secret s according to the LSSS matrix, we have: $\sum_i \omega_i \lambda_i = s$. Essentially, a user will be able to decrypt a ciphertext with access matrix A if and only if the rows of A labeled by the user's attributes include the vector $(1, 0, \ldots, 0)$ in their span.

Now, we 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.

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

Setup(λ, 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$(PK, CT, SK) \rightarrow M$*.* 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.

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 1. *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 [5,25,39], we do not impose this restriction on the attacker.

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 $N = p_1p_2p_3$, and $e : G^2 \to G_T$ is a non-degenerate bilinear map.

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 [28]. 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. The assumptions were proven to be generically secure in [28].

In the assumptions below, we let $G_{p_1p_2}$, e.g., denote the subgroup of order p_1p_2 in G.

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

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

$$g \xleftarrow{R} G_{p_1},\ X_3 \xleftarrow{R} G_{p_3},$$
$$D = (\mathbb{G}, g, X_3),$$
$$T_1 \xleftarrow{R} G_{p_1p_2},\ T_2 \xleftarrow{R} G_{p_1}.$$

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 2. *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:

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

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

$$D = (\mathbb{G}, g, X_1X_2, X_3, Y_2Y_3),$$

$$T_1 \xleftarrow{R} G,\ T_2 \xleftarrow{R} G_{p_1p_3}.$$

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 $G_{p_1p_3}$ to denote the subgroup of order p_1p_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 3. *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:

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

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

$$D = (\mathbb{G}, g, g^{\alpha}X_2, X_3, g^sY_2, Z_2),$$

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

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 4. *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 $N = p_1 p_2 p_3$ with LSSS access structures. We note the strong resemblance between our system and the selectively secure CP-ABE system of Waters [39]. The KP-ABE system we give in the full version of this paper also bears a strong resemblance to the selectively secure schemes in [25]. We thus provide additional examples of the phenomenon noted by [40,28]: 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 [40,28].

Construction

Setup$(\lambda, U) \rightarrow PK, MSK$. 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 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):

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

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

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 $C/e(g,g)^{\alpha s}$.

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. We give the proof of the most interesting lemma below, and the rest of the proofs can be found in the full version of this paper. For notational purposes in the lemmas below, we think of $Game_{0,2}$ as another way of denoting Game 0.

Lemma 1. *Suppose there is an efficient algorithm $\mathcal{A}$ such that $Game_{Real}Adv_{\mathcal{A}} - Game_0 Adv_{\mathcal{A}} = \epsilon$. Then we can construct an efficient algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 1.*

Lemma 2. *Suppose there is an efficient algorithm $\mathcal{A}$ such that $Game_{k-1,2}Adv_{\mathcal{A}} - Game_{k,1}Adv_{\mathcal{A}} = \epsilon$. Then we can construct an efficient 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 Game$_{k-1,2}$ or 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 $(1, 0, \ldots, 0)$. So for shares $\delta_x = A^*_x \cdot u$ in the G_{p_2} subgroup, we can write $u = u_R + u_W$, where u_R is in the space R and u_W is in its orthogonal complement, W. We note that $u_1 = u \cdot (1, 0, \ldots, 0)$ cannot be determined from u_R alone - some information about u_W is needed.

The only places u_W 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 u_W 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 $T \in G_{p_1p_3}$, then $\mathcal{B}$ has properly simulated 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 Game$_{k,1}$. $\mathcal{B}$ can therefore use the output of $\mathcal{A}$ to gain advantage negligibly close to ϵ in breaking Assumption 2.

Lemma 3. *Suppose there is an efficient algorithm $\mathcal{A}$ such that $Game_{k,1}Adv_{\mathcal{A}} - Game_{k,2}Adv_{\mathcal{A}} = \epsilon$. Then we can construct an efficient algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 2.*

Lemma 4. *Suppose there is an efficient algorithm $\mathcal{A}$ such that $Game_{q,2}Adv_{\mathcal{A}} - Game_{Final}Adv_{\mathcal{A}} = \epsilon$. Then we can construct an efficient algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 3.*

We have now proven the following theorem:

Theorem 1. *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.

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 note that the public parameters and key sizes will grow by a factor of k, but the encoding does not increase the size of the ciphertext.

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 the full version of this paper. 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 $H : \{0,1\}^* \rightarrow \mathbb{Z}_{p_1}^*$. 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 $T : \mathbb{Z}_{p_1} \rightarrow G_{p_1}$ based on a degree n polynomial. Goyal, Pandey, Sahai, and Waters [25] 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. Waters [40] 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 [28] 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 [20] 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

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

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, $X := (\chi_{i,j}) \overset{\mathsf{U}}{\leftarrow} GL(N, \mathbb{F}_q)$, 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 $(x_1, \ldots, x_N)_{\mathbb{B}} = (\vec{x})_{\mathbb{B}} := \boldsymbol{x}$ and $(y_1, \ldots, y_N)_{\mathbb{B}^*} = (\vec{y})_{\mathbb{B}^*} := \boldsymbol{y}$, and $e(\boldsymbol{x}, \boldsymbol{y}) = e(g, g)^{\sum_{i=1}^{N} x_i y_i} = g_T^{\vec{x} \cdot \vec{y}} \in \mathbb{G}_T$.

Intractable problem: One of the most natural decisional problems in this approach is the decisional subspace problem [30]. 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 $(v_1, \ldots, v_{N_1}) \overset{\mathsf{U}}{\leftarrow} \mathbb{F}_q^{N_1}$ 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.

Dual System Encryption Methodology. At the top level of strategy of the security proof, we follow the dual system encryption methodology proposed by Waters [40]. 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 [28,40].

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 [28] 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.

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, $A(x) \rightarrow a$ (e.g., $A(x) \rightarrow 1$) denotes the event that machine (algorithm) A outputs a on input x. A function $f : \mathbb{N} \rightarrow \mathbb{R}$ 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 5. *"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 $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ 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 [30,31] constructed by using symmetric bilinear pairing groups given in Definition 5.

Definition 6. *"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 $e : \mathbb{V} \times \mathbb{V} \rightarrow \mathbb{G}_T$.*

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 the full version of this paper. 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 [30].)

We describe random dual orthonormal bases generator $\mathcal{G}_{\mathsf{ob}}$ below, which is used as a subroutine 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 7 (n-eDDH: n-Extended Decisional Diffie-Hellman Assumption). *The n-eDDH problem is to guess $\beta \in \{0,1\}$, given $(\mathsf{param}_{\mathbb{G}},\ g, g^\kappa, \{g^{\omega + \gamma_i h_i}, g^{\gamma_i}, g^{h_i}\}_{1 \le i \le n},\ \{g^{\gamma_i h_j}\}_{1 \le i \neq j \le n},\ Y_\beta) \xleftarrow{\mathsf{R}} \mathcal{G}_\beta^{n\text{-}\mathsf{eDDH}}(1^\lambda)$, where*

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

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:

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

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

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

Lemma 5. *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 5 is given in the full version of this paper.

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 8. *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 $c \overset{\mathsf{R}}{\leftarrow} \mathsf{Enc}(\mathsf{pk}, m, \overrightarrow{x})$, 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 9. *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* $\mathsf{sk}_{\overrightarrow{v}} \overset{\mathsf{R}}{\leftarrow} \mathsf{KeyGen}(\mathsf{sk}, \overrightarrow{v})$.
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 $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{PE,AH}}(\lambda) := \mathsf{Pr}[b' = b] - 1/2$.

Remark: In Definition 9, 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 [27], 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

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 $\overrightarrow{x} \cdot \overrightarrow{v} = 0$.

Security

Theorem 2. *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,*

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

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

We will show Lemmas 6, 7, and 8 for the proof of Theorem 2. The proofs of these lemmas are given in the full version of this paper.

Definition 10. *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 6. *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, $\mathsf{Adv}_{\mathcal{C}}^{n\text{-}\mathsf{eDDH}}(\lambda) = \mathsf{Adv}_{\mathcal{B}}^{\mathsf{P1}}(\lambda)$.*

Definition 11. *Problem 2 is to guess $\beta \in \{0,1\}$, given $(\mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \{\boldsymbol{h}_{\beta,i}^*, \boldsymbol{e}_i\}_{i=1,\ldots,n}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\beta}^{\mathsf{P2}}(1^\lambda, n)$, where*

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

for $\beta \xleftarrow{\mathsf{U}} \{0,1\}$. For a probabilistic machine $\mathcal{B}$, the advantage of $\mathcal{B}$ for Problem 2, $\mathsf{Adv}_{\mathcal{B}}^{\mathsf{P2}}(\lambda)$, is similarly defined as in Definition 10.

Lemma 7. *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, $\mathsf{Adv}^{n\text{-}\mathsf{eDDH}}_{\mathcal{C}}(\lambda) = \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}}(\lambda)$.*

Lemma 8. *Let $C := \{(\vec{x}, \vec{v}) | \vec{x} \cdot \vec{v} \neq 0\} \subset V \times V^*$ where V is n-dimensional vector space $\mathbb{F}_q^n$, and V^* its dual. For all $(\vec{x}, \vec{v}) \in C$, for all $(\vec{r}, \vec{w}) \in C$,*

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

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

Proof Outline of Theorem 2. 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 $\boldsymbol{k}^{*\,\mathsf{norm}}_{\vec{v}}$ (with predicate vector $\vec{v}$) is a correct form of the secret key of the proposed PE scheme, i.e., $\boldsymbol{k}^{*\,\mathsf{norm}}_{\vec{v}} := \sigma(\sum_{i=1}^{n} v_i \boldsymbol{b}^*_i) + \boldsymbol{b}^*_{2n+1} + \eta \boldsymbol{b}^*_{2n+2} = (\sigma\vec{v}, \vec{0}_n, 1, \eta, 0)_{\mathbb{B}^*}$, where $\vec{0}_n := (\overbrace{0, \cdots, 0}^{n})$. Similarly, a *normal* ciphertext (with attribute $\vec{x}$) is $(\boldsymbol{c}^{\mathsf{norm}}_{\vec{x}}, c_2)$ with $\boldsymbol{c}^{\mathsf{norm}}_{\vec{x}} := \delta_1(\sum_{i=1}^{n} x_i \boldsymbol{b}_i) + \zeta \boldsymbol{b}_{2n+1} + \delta_2 \boldsymbol{b}_{2n+3} = (\delta_1 \vec{x}, \vec{0}_n, \zeta, 0, \delta_2)_{\mathbb{B}}$. (Hereafter we will ignore c_2 since c_2 is always correctly generated.) A *semi-functional* secret key is $\boldsymbol{k}^{*\,\mathsf{semi}}_{\vec{v}} := (\sigma\vec{v}, \vec{r}, 1, \eta, 0)_{\mathbb{B}^*}$ and a *semi-functional* ciphertext is $\boldsymbol{c}^{\mathsf{semi}}_{\vec{x}} := (\delta_1\vec{x}, \vec{s}, \zeta, 0, \delta_2)_{\mathbb{B}}$, where $\vec{r}, \vec{s} \xleftarrow{\mathsf{U}} \mathbb{F}_q^n$. If $\vec{x} \cdot \vec{v} = 0$, then $e(\boldsymbol{c}^{\mathsf{norm}}_{\vec{x}}, \boldsymbol{k}^{*\,\mathsf{norm}}_{\vec{v}}) = e(\boldsymbol{c}^{\mathsf{norm}}_{\vec{x}}, \boldsymbol{k}^{*\,\mathsf{semi}}_{\vec{v}}) = e(\boldsymbol{c}^{\mathsf{semi}}_{\vec{x}}, \boldsymbol{k}^{*\,\mathsf{norm}}_{\vec{v}}) = g_T^{\zeta}$, which leads to correct decryption. In contrast, $e(\boldsymbol{c}^{\mathsf{semi}}_{\vec{x}}, \boldsymbol{k}^{*\,\mathsf{semi}}_{\vec{v}}) = g_T^{\vec{s}\cdot\vec{r}+\zeta}$, which is uniformly and independently distributed over $\mathbb{F}_q$ since $\vec{r}, \vec{s} \xleftarrow{\mathsf{U}} \mathbb{F}_q^n$, (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 9). The advantage of Problem 1 is proven to be equivalent to that of the n-eDDH assumption (Lemma 6).

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 7 and 10).

Problem 2 is based on our key trick (explained in Section 3.1). Here, we introduce *special form of semi-functional* keys and ciphertexts such that $\boldsymbol{k}^{*\ \mathsf{spec.semi}}_{\vec{v}} := (\sigma\vec{v}, (\tau\vec{v}Z), 1, \eta, 0)_{\mathbb{B}^*}$, and $\boldsymbol{c}^{\mathsf{spec.semi}}_{\vec{x}} := (\delta\vec{x}, (\rho\vec{x}U), \zeta, 0, \delta_2)_{\mathbb{B}}$, where Z is a random regular $(n \times n)$-matrix, $U := (Z^{-1})^{\mathrm{T}}$, and $\tau, \rho \xleftarrow{\mathsf{U}} \mathbb{F}_q$.

$\boldsymbol{k}^{*\ \mathsf{spec.semi}}_{\vec{v}}$ can decrypt $\boldsymbol{c}^{\mathsf{spec.semi}}_{\vec{x}}$ for *all* vectors $\vec{x}$ with $\vec{v}\cdot\vec{x} = 0$, since $(\tau\vec{v}Z)\cdot(\rho\vec{x}U) = \tau\rho(\vec{v}\cdot\vec{x})$, i.e., $e(\boldsymbol{c}^{\mathsf{spec.semi}}_{\vec{x}}, \boldsymbol{k}^{*\ \mathsf{spec.semi}}_{\vec{v}}) = g^{(\delta_1\sigma+\tau\rho)(\vec{v}\cdot\vec{x})+\zeta}$. In addition, $(\tau\vec{v}Z)$ and $(\rho\vec{x}U)$ are uniformly and pairwise-independently distributed (i.e., equivalently distributed to $(\vec{r}, \vec{s}) \xleftarrow{\mathsf{U}} (\mathbb{F}_q^n)^2 \backslash \{(\vec{r}, \vec{s}) \mid \vec{r}\cdot\vec{s} = 0\}$), when $\vec{v}\cdot\vec{x} \neq 0$ (Lemma 8). Therefore, the joint distribution of $\boldsymbol{k}^{*\ \mathsf{spec.semi}}_{\vec{v}}$ and $\boldsymbol{c}^{\mathsf{spec.semi}}_{\vec{x}}$ is equivalent to that of an independent pair of $\boldsymbol{k}^{*\ \mathsf{semi}}_{\vec{v}}$ and $\boldsymbol{c}^{\mathsf{semi}}_{\vec{x}}$ (except with probability $1/q$), when $\vec{v}\cdot\vec{x} \neq 0$.

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 11).

Proof of Theorem 2: To prove Theorem 2, 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 $(\vec{x}^{(0)}, \vec{x}^{(1)})$ 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 $\delta_1, \delta_2, \zeta \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $b \xleftarrow{\mathsf{U}} \{0,1\}$, $(x_1^{(b)}, \ldots, x_n^{(b)}) := \vec{x}^{(b)}$, and $(w_1, \ldots, w_n) \xleftarrow{\mathsf{U}} \mathbb{F}_q^n \setminus \{\vec{0}\}$.

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 $\vec{v} := (v_1, \ldots, v_n)$ 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 $\vec{r} := (r_1, \ldots, r_n) \xleftarrow{\mathsf{U}} \mathbb{F}_q^n$.

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 $(\vec{x}^{(0)}, \vec{x}^{(1)})$ is

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

where $x_1', \ldots, x_n', \delta_2, \zeta, \zeta' \xleftarrow{\mathsf{U}} \mathbb{F}_q, b \xleftarrow{\mathsf{U}} \{0,1\}$, and $(w_1, \ldots, w_n) \xleftarrow{\mathsf{U}} \mathbb{F}_q^n \setminus \{\vec{0}\}$. In particular, we note that $(x_1', \ldots, x_n')$ and ζ' are chosen uniformly and independently from $\vec{x}^{(0)}, \vec{x}^{(1)}$ 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 12.

We will use three lemmas (Lemmas 9, 10, 11) that evaluate the gaps between pairs of $\mathsf{Adv}^{(0)}_{\mathcal{A}}(\lambda), \mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda), \mathsf{Adv}^{(2\text{-}k)}_{\mathcal{A}}(\lambda)$ $(k = 1, \ldots, \nu), \mathsf{Adv}^{(3)}_{\mathcal{A}}(\lambda)$. From these lemmas, we obtain $\mathsf{Adv}^{\mathsf{PE,AH}}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{(0)}_{\mathcal{A}}(\lambda) \leq \left|\mathsf{Adv}^{(0)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda)\right| + \sum_{k=1}^{\nu} \left|\mathsf{Adv}^{(2\text{-}(k-1))}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(2\text{-}k)}_{\mathcal{A}}(\lambda)\right| + \left|\mathsf{Adv}^{(2\text{-}\nu)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(3)}_{\mathcal{A}}(\lambda)\right| + \mathsf{Adv}^{(3)}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}_0}(\lambda) + \sum_{k=1}^{\nu} \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}_k}(\lambda) + \frac{\nu}{q}$. From Lemmas 6 and 7, 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 $\mathsf{Adv}^{n\text{-}\mathsf{eDDH}}_{\mathcal{C}_0}(\lambda) = \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}_0}(\lambda)$ and $\mathsf{Adv}^{n\text{-}\mathsf{eDDH}}_{\mathcal{C}_k}(\lambda) = \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}_k}(\lambda)$ $(k = 1, \ldots, \nu)$. Hence, $\mathsf{Adv}^{\mathsf{PE,AH}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}_0}(\lambda) + \sum_{k=1}^{\nu} \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}_k}(\lambda) + \frac{\nu}{q} \leq \sum_{k=0}^{\nu} \mathsf{Adv}^{n\text{-}\mathsf{eDDH}}_{\mathcal{C}_k}(\lambda) + \frac{\nu}{q}$. This completes the proof of Theorem 2. □

The proofs of the following lemmas appear in the full version of this paper.

Lemma 9. *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, $|\mathsf{Adv}^{(0)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda)| = \mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}_0}(\lambda)$.*

Lemma 10. *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, $|\mathsf{Adv}^{(2\text{-}(k-1))}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(2\text{-}k)}_{\mathcal{A}}(\lambda)| \leq \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}_k}(\lambda) + \frac{1}{q}$.*

Lemma 11. *For any adversary $\mathcal{A}$, $\mathsf{Adv}^{(2\text{-}\nu)}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{(3)}_{\mathcal{A}}(\lambda)$.*

Lemma 12. *For any adversary $\mathcal{A}$, $\mathsf{Adv}^{(3)}_{\mathcal{A}}(\lambda) = 0$.*

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 the full version of this paper.

Construction

```
$\mathsf{Setup}(1^\lambda, \vec{\mu} := (n, d; \mu_1, \ldots, \mu_d))$ :  $(\mathsf{param}_{\mathbb{V}}, \mathbb{B}, \mathbb{B}^*) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^\lambda, 2n+3)$,
   $\widehat{\mathbb{B}} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n, \boldsymbol{b}_{2n+1}, \boldsymbol{b}_{2n+3})$,   $\mathsf{sk} := \mathbb{B}^*$,   $\mathsf{pk} := (1^\lambda, \mathsf{param}_{\mathbb{V}}, \widehat{\mathbb{B}})$,
   return  $\mathsf{sk}$, $\mathsf{pk}$.
$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{sk}, (\vec{v}_1, \ldots, \vec{v}_\ell) := ((v_1, \ldots, v_{\mu_1}), \ldots, (v_{\mu_{\ell-1}+1}, \ldots, v_{\mu_\ell}))$ :
   $\sigma_{\mathsf{dec},t}, \eta_{\mathsf{dec}}, \; \sigma_{\mathsf{ran},j,t}, \eta_{\mathsf{ran},j} \; (j = 1, .., \ell+1), \; \sigma_{\mathsf{del},j,t}, \eta_{\mathsf{del},j} \; (j = 1, .., n), \; \psi \xleftarrow{\mathsf{U}} \mathbb{F}_q$
         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  $\vec{\boldsymbol{k}}^*_\ell := (\boldsymbol{k}^*_{\ell,\mathsf{dec}}, \boldsymbol{k}^*_{\ell,\mathsf{ran},1}, \ldots, \boldsymbol{k}^*_{\ell,\mathsf{ran},\ell+1}, \boldsymbol{k}^*_{\ell,\mathsf{del},\mu_\ell+1}, \ldots, \boldsymbol{k}^*_{\ell,\mathsf{del},n})$.
```

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

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

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

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 the full version of this paper.

Security

Theorem 3. *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,*

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

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

The proof is given in the full version of this paper.

References

1. Al-Riyami, S., Malone-Lee, J., Smart, N.: Escrow-free encryption supporting cryptographic workflow. Int. J. Inf. Sec. 5, 217–229 (2006)
2. Bagga, W., Molva, R., Crosta, S.: Policy-based encryption schemes from bilinear pairings. In: ASIACCS, p. 368 (2006)
3. Barbosa, M., Farshim, P.: Secure cryptographic workflow in the standarad model. In: Barua, R., Lange, T. (eds.) INDOCRYPT 2006. LNCS, vol. 4329, pp. 379–393. Springer, Heidelberg (2006)

4. Beimel, A.: Secure schemes for secret sharing and key distribution. PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel (1996)
5. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: Proceedings of the IEEE Symposium on Security and Privacy (2007)
6. Boneh, D., Boyen, X.: Efficient selective-id secure identity based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
7. Boneh, D., Boyen, X.: Secure identity based encryption without random oracles. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 443–459. Springer, Heidelberg (2004)
8. Boneh, D., Boyen, X., Goh, E.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005)
9. Boneh, D., Franklin, M.: Identity based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
10. Boneh, D., Goh, E., Nissim, K.: Evaluating 2-dnf formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–342. Springer, Heidelberg (2005)
11. Boneh, D., Katz, J.: Improved efficiency for cca-secure cryptosystems built using identity based encryption. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 87–103. Springer, Heidelberg (2005)
12. Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 535–554. Springer, Heidelberg (2007)
13. Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (without random oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006)
14. Bradshaw, R., Holt, J., Seamons, K.: Concealing complex policies with hidden credentials. In: CCS, pp. 146–157 (2004)
15. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 255–271. Springer, Heidelberg (2003)
16. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
17. Chase, M.: Multi-authority attribute based encryption. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 515–534. Springer, Heidelberg (2007)
18. Cheung, L., Newport, C.: Provably secure ciphertext policy abe. In: CCS, pp. 456–465 (2007)
19. Cocks, C.: An identity based encryption scheme based on quadratic residues. In: Honary, B. (ed.) Cryptography and Coding 2001. LNCS, vol. 2260, pp. 26–28. Springer, Heidelberg (2001)
20. Freeman, D.M.: Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In: EUROCRYPT (2010)
21. Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)
22. Gentry, C., Halevi, S.: Hierarchical identity based encryption with polynomially many levels. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 437–456. Springer, Heidelberg (2009)

23. Gentry, C., Silverberg, A.: Hierarchical id-based cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg (2002)
24. Goyal, V., Jain, A., Pandey, O., Sahai, A.: Bounded ciphertext policy attribute-based encryption. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 579–591. Springer, Heidelberg (2008)
25. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute Based Encryption for Fine-Grained Access Conrol of Encrypted Data. In: CCS (2006)
26. Horwitz, J., Lynn, B.: Toward hierarchical identity-based encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 466–481. Springer, Heidelberg (2002)
27. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (2008)
28. Lewko, A., Waters, B.: New techniques for dual system encryption and fully secure hibe with short ciphertexts. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 455–479. Springer, Heidelberg (2010)
29. Miklau, G., Suciu, D.: Controlling access to published data using cryptography. In: VLDB, pp. 898–909 (2003)
30. Okamoto, T., Takashima, K.: Homomorphic encryption and signatures from vector decomposition. In: Galbraith, S.D., Paterson, K.G. (eds.) Pairing 2008. LNCS, vol. 5209, pp. 57–74. Springer, Heidelberg (2008)
31. Okamoto, T., Takashima, K.: Hierarchical predicate encryption for inner-products. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 214–231. Springer, Heidelberg (2009)
32. Ostrovksy, R., Sahai, A., Waters, B.: Attribute Based Encryption with Non-Monotonic Access Structures. In: CCS (2007)
33. Pirretti, M., Traynor, P., McDaniel, P., Waters, B.: Secure attribute-based systems. In: CCS, pp. 99–112 (2006)
34. Sahai, A., Waters, B.: Fuzzy Identity Based Encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
35. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
36. Shi, E., Waters, B.: Delegating capabilities in predicate encryption systems. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 560–578. Springer, Heidelberg (2008)
37. Smart, N.: Access control using pairing based cryptography. In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp. 111–121. Springer, Heidelberg (2003)
38. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
39. Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. Cryptology ePrint Archive, Report 2008/290 (2008)
40. Waters, B.: Dual system encryption: realizing fully secure ibe and hibe under simple assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)