Hindawi
Security and Communication Networks
Volume 2021, Article ID 6678194, 12 pages
https://doi.org/10.1155/2021/6678194

Research Article

Private Predicate Encryption for Inner Product from Key-Homomorphic Pseudorandom Function

Yi-Fan Tseng, Zi-Yuan Liu, Jen-Chieh Hsu, and Raylin Tso

Department of Computer Science, National Chengchi University, Taipei 11605, Taiwan

Correspondence should be addressed to Zi-Yuan Liu; zyliu@cs.nccu.edu.tw

Received 4 November 2020; Revised 7 December 2020; Accepted 21 January 2021; Published 12 February 2021

Academic Editor: David Megías

Predicate encryption (PE), formalized by Katz et al., is a new paradigm of public-key encryption that conceptually captures the public-key encryption that supports fine-grained access control policy. Because of the nature of PE, it is used for cloud storage so that users can retrieve encrypted data without revealing any information about the data to cloud servers and other users. Although lots of PE schemes have been studied, the predicate-hiding security is seldom considered; that is, the user's secret key may leak sensitive information of the predicate. Additionally, the security of the current predicate-hiding PE schemes relies on the discrete logarithm assumption which cannot resist the quantum attacks in the future. In this paper, we propose a generic PE for inner product under symmetric-key setting, called private IPE, from specific key-homomorphic pseudorandom function (PRF). The rigorous proofs are provided to show that the construction is payload-hiding, attribute-hiding, and predicate-hiding secure. With the advantage of the generic construction, if the underlying PRF can resist quantum attacks, then, through our proposed generic construction, a quantum-resistant private IPE can be obtained.

1. Introduction

In recent years, cloud computing has become increasingly important as smartphones and Internet of Things devices are widely used in our life. Users typically upload their data to the cloud to achieve efficient computing and reduce storage requirements of their devices. Due to the fact that the uploaded data are sensitive, users may consider using authentication protocol [1–4] and encryption schemes [5, 6] to protect their data privacy in cloud environment. One novel approach is to encrypt data before it is uploaded to the cloud. However, encrypted data loses flexibility in data usage, such as fine-grained control over access to encrypted data. For example, a user may want to search for and download ciphertext that corresponds to certain attributes. If each piece of data is purely encrypted, the only way is to download all the ciphertexts and decrypt them for search. Unfortunately, this approach would be very inefficient. Therefore, how to efficiently control the access to encrypted data and ensure the privacy and security of data is an urgent issue for cloud computing.

Predicate encryption (PE) [7], formalized by Katz et al., is a general paradigm that conceptually captures the public-key encryption supporting fine-grained access control policy. In a PE scheme for a predicate function P, a secret key, issued by a trusted authority, is associated with a key attribute $\mathbf{y}$, while the ciphertext is associated with a ciphertext attribute $\mathbf{x}$. Specifically, the ciphertext can be decrypted using the secret key if and only if $P(\mathbf{x},\mathbf{y}) = 1$. Therefore, PE can be used as access control mechanism for the previous cloud storage scenario and provide the flexibility for encryption schemes, which allows sender to encrypt data with more complicated access policy. For example, in a school scenario, the secret keys of each teacher and each student are associated with key attributes "teacher" and "student," respectively. If the principal wants to encrypt a file that can only be decrypted by each student and teacher, he/she can use a PE supporting "belong to" functionality and encrypt this file with a ciphertext attribute "student or teacher." Because the key attributes "teacher" and "student" belong to ciphertext attribute “student or teacher,” the secret keys associated with these key attributes can decrypt the ciphertext.

Additionally, Katz et al. proposed the first PE supporting inner product predicate, called PE for inner product (IPE), whereas ciphertext can be decrypted if and only if the inner product of $\mathbf{x}$ and $\mathbf{y}$ is equal to 0. They further suggested that IPE can be used to build other more flexible schemes, such as (anonymous) identity-based encryption [8], hidden vector encryption [9, 10], CNF/DNF formulas [7], PE schemes supporting polynomial evaluation [11], and exact thresholds [12]. The most basic security requirement of IPE, called payload-hiding, stipulates that a ciphertext does not reveal any information of the plaintext if $P(\mathbf{x}, \mathbf{y}) = 1$. A stronger security requirement of PE is attribute-hiding, which stipulates that a ciphertext reveals nothing about the ciphertext attribute. Although a lot of attribute-hiding IPE schemes [13–16] have been studied, seldom schemes [17–19] focus on the predicate-hiding security. In more detail, a secret key may reveal some sensitive information of the predicate that belongs to the key holder. Actually, in public-key cryptosystem, since the encryption algorithm is publicly accessible, any user can adaptively generate a ciphertext. The user who has obtained a secret key can evaluate its predicate with possible ciphertexts; thus it is hard to achieve predicate-hiding in the public-key setting.

Shen et al. [18] first considered constructing the IPE under symmetric-key setting, a.k.a. private IPE, to achieve predicate-hiding security requirement. More precisely, in the work, when generating a secret key, generating a ciphertext requires a master secret key, so that not every user can adaptively generate a ciphertext to test which predicate is embedded in the secret key. Compared with IPE under public-key setting, private IPE is more suitable for cloud storage under self-use scenario. For example, as shown in Figure 1, Alice uses the cloud storage service to store her files. For privacy concern, she uses private IPE as an access control mechanism. Alice not only uploads an encrypted file $ct_{File,i}$ but also uploads another ciphertext $ct_{\mathbf{x},i} = Encrypt(SK, \mathbf{x}, M = 1)$ for a specific ciphertext attribute by using private IPE. When Alice wants to retrieve encrypted files, she can send the secret key for some key attribute, that is, $sk_{\mathbf{y}} \longleftarrow KeyGen(SK, \mathbf{y})$, to the cloud. The cloud can then evaluate the predicate on each ciphertext by performing decryption. If the predicate is satisfied, that is, $1 \overset{?}{=} Decrypt(ct_{\mathbf{x},i}, sk_{\mathbf{y}})$, the cloud returns the corresponding encrypted files of those ciphertexts.

After Shen et al.’s pioneering work [18], Yoshino et al. [19] provided a more practical IPE scheme that uses only three groups, whereas [18] required four groups. In addition, Kawai and Takashima [17] then introduced a predicate-hiding IPE, where the security is proven under the decision linear assumption without random oracles. However, the sizes of the secret keys of the above schemes [17–19] are linearly related to the lengths of the key attributes. Due to the fact that users may obtain many secret keys for decrypting different ciphertext, it is important to reduce the key size of secret key. In addition, Shor [20, 21] has shown that existing quantum algorithms can break the discrete logarithm and factoring assumptions. Therefore, the current private IPE schemes [17–19] are susceptible to quantum attack. Hence, how to construct a quantum-resistant private IPE scheme where the secret key is of constant size remains an open issue.

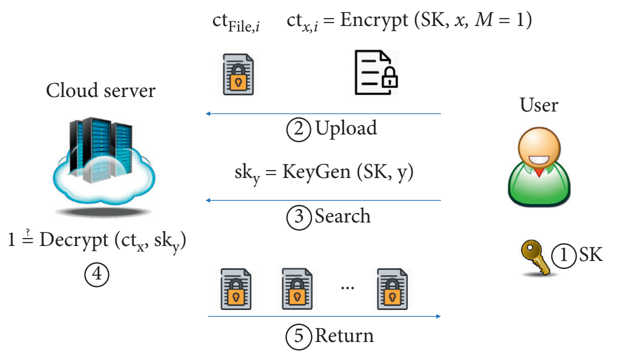
$ct_{File,i}$
$ct_{x,i}$ = Encrypt (SK, x, M = 1)
Cloud server
User
② Upload
sk_y = KeyGen (SK, y)
③ Search
$1 \overset{?}{=}$ Decrypt (ct_x, sk_y)
④
①SK
...
⑤Return

FIGURE 1: Private IPE scheme for cloud storage in self-use scenario.

1.1. Our Contributions. In this paper, inspired by Alamati et al.’s work [22], we propose a generic private IPE construction by utilizing specific key-homomorphic pseudorandom functions (PRF). By the advantage of the generic construction, the construction enjoys the security properties of the underlying primitives. Therefore, if the underlying key-homomorphic PRF is quantum-resistant, we further obtain a quantum-resistant private IPE scheme. In particular, in our construction, we require the underlying key-homomorphic PRF to have the following property for decryption correctness: the key space $\mathscr{K}$ and the output space $\mathscr{Y}$ are equal to $\mathbb{Z}_q$, for some prime q.

To obtain a private IPE scheme with constant-size secret key, we carefully use the key-homomorphic property of the key-homomorphic PRF to map each predicate attribute to the inner product of master secret key and secret key. That is, $sk_{\mathbf{y}} = \sum_{i=1}^{\ell} (\sum_{j=1}^{y_i} F(a_i, h)) = F(\langle \mathbf{a}, \mathbf{y} \rangle, h)$, where $\mathbf{y} = (y_1, \ldots, y_\ell)$ is a predicate vector and $(\mathbf{a} = (a_1, \ldots, a_\ell), h)$ is the master secret key. Hence, the size of secret key is only $\log_2 q$, where q is the underlying modulo.

In addition, the rigorous security proofs are provided to demonstrate that if the underlying key-homomorphic PRF satisfies pseudorandomness (i.e., the output value of key-homomorphic PRF is indistinguishable from the value randomly chosen from $\mathscr{Y}$), the proposed construction satisfies the criteria of payload-hiding, attribute-hiding, and predicate-hiding privacy. The comparison of our construction with other state-of-the-art private IPE schemes is presented to show that our result is not only more secure but also more efficient with respect to the size of secret key.

In summary, this work introduces a generic construction to show how to obtain the first quantum-resistant private IPE scheme with a constant-size secret key.

1.2. Paper Organization. The rest of the paper is organized as follows. Section 2 recalls the definition of the PRF used in our generic construction. Moreover, Section 3 provides the definition and security requirement of the private PE. Next, Sections 4 and Section 5 introduce and provide the security proofs of our generic constriction, respectively. Section 6 compares our proposed construction with the related private IPE schemes. Finally, Section 7 concludes this study.

2. Pseudorandom Function (PRF)

In this section, we recall the definition of pseudorandom function from [23].

Definition 1 (pseudorandom functions [23]). A PRF F: $\mathscr{K} \times \mathscr{X} \longrightarrow \mathscr{Y}$ is a keyed function defined over a key space $\mathscr{K}$, a domain $\mathscr{X}$, and a range $\mathscr{Y}$ (these sets may be parameterized by the security parameter λ), whose output is indistinguishable from a truly random value. The security of a PRF can be defined by the two experiments $EXP(0)$ and $EXP(1)$ with an adversary $\mathscr{A}$. At first, a key k is uniformly randomly chosen from the key space $\mathscr{K}$. Given the description of the PRF, the adversary is then allowed to make queries to the following oracles:

(i) *Evaluate.* Given $x \in \mathscr{X}$ from $\mathscr{A}$, the oracle returns $F(k, x)$ to $\mathscr{A}$.

(ii) *Challenge.* Given $x \in \mathscr{X}$ from $\mathscr{A}$, where x has not been queried to evaluate Oracle, if $b = 1$, then the oracle returns $F(k, x)$, and if $b = 0$, then the oracle returns a random $y \overset{\$}{\longleftarrow} \mathscr{Y}$.

Once the adversary is done querying the oracles, it outputs a bit $b' \in \{0, 1\}$. For $b = 0, 1$, we define W_b as the event where the adversary outputs $b' = 1$ in the experiment $EXP(b)$. The advantage of an adversary $\mathscr{A}$ is defined as

$$\mathrm{Adv}_{\mathscr{A}}^{\mathrm{PRF}}\left(1^{\lambda}\right) = \left|\Pr\left[W_1\right] - \Pr\left[W_0\right]\right|. \quad (1)$$

We say that a PRF is secure if, for all PPT adversary $\mathscr{A}$, $\mathrm{Adv}_{\mathscr{A}}^{\mathrm{PRF}}(1^{\lambda})$ is negligible.

Definition 2 (key-homomorphic PRF [24]). Let $(\mathscr{K}, \circ)$ and $(\mathscr{Y}, *)$ be groups. Then, a keyed function F: $\mathscr{K} \times \mathscr{X} \longrightarrow \mathscr{Y}$ is a key-homomorphic PRF:

(i) F is a secure PRF.

(ii) For every $k_1, k_2 \in \mathscr{K}$ and every input $x \in \mathscr{X}$, we have

$$F\left(k_1, x\right) * F\left(k_2, x\right) = F\left(k_1 \circ k_2, x\right). \quad (2)$$

Definition 3 (pseudorandom generators [24]). A pseudorandom generator (PRG) is an efficiently computable function G: $\mathscr{X} \longrightarrow \mathscr{Y}$ with the following security, where $(\mathscr{X}, \circ)$ and $(\mathscr{Y}, *)$ are groups. The security of a PRG is secure if, for any PPT algorithm $\mathscr{A}$, is negligible.

$$\mathrm{Adv}_{\mathscr{A}}^{\mathrm{PRG}}\left(1^{\lambda}\right) = |\Pr[\mathscr{A}(G(x)) = 1; x \longleftarrow \mathscr{X}] - \Pr[\mathscr{A}(R) = 1; R \overset{\$}{\longleftarrow} \mathscr{Y}]|. \quad (3)$$

3. Private Predicate Encryption

Let $\{P = P_{\ell}\}_{\ell \in \mathbb{N}^c}$ for some constant $c \in \mathbb{N}$ be a predicate family, where P_{ℓ}: $\mathfrak{A}_{\ell} \times \mathfrak{P}_{\ell} \longrightarrow \{0, 1\}$ is a predicate function defined over a ciphertext attribute space $\mathfrak{A}_{\ell}$ and a key attribute space $\mathfrak{P}_{\ell}$. The family index ℓ specifies the description of a predicate from the family. We would occasionally omit the index ℓ when the context is clear.

3.1. System Model. A private PE for predicate function P: $\mathfrak{A} \times \mathfrak{P} \longrightarrow \{0, 1\}$ consists of four algorithms: *Setup*, *KeyGen*, *Encrypt*, and *Decrypt*. The details of the algorithms are shown as follows:

(i) $Setup(1^{\lambda}, 1^{\ell}) \longrightarrow (pp, SK)$. Given the security parameters and the family index (λ, ℓ), the algorithm outputs the system parameter pp and the secret key SK. Note that the description of $\mathfrak{A}$ and $\mathfrak{P}$ will be implicitly included in pp.

(ii) $Encrypt(pp, SK, \mathbf{x}, M) \longrightarrow ct_{\mathbf{x}}$. Given the system parameter pp, a secret key SK, a ciphertext attribute $\mathbf{x} \in \mathfrak{A}$, and a message M, the algorithm outputs a ciphertext $ct_{\mathbf{x}}$ for $\mathbf{x}$.

(iii) $KeyGen(pp, SK, \mathbf{y}) \longrightarrow sk_{\mathbf{y}}$. Given the system parameter pp, a secret key SK, and a key attribute $\mathbf{y} \in \mathfrak{P}$, the algorithm outputs the secret key $sk_{\mathbf{y}}$ for $\mathbf{y}$.

(iv) $Decrypt(pp, ct_{\mathbf{x}}, sk_{\mathbf{y}}) \longrightarrow (M/\perp)$. Given the system parameter pp, a ciphertext $ct_{\mathbf{x}}$, and a secret key $sk_{\mathbf{y}}$, the algorithm outputs a message M or an error symbol $\perp$.

Definition 4 (correctness). For all $\lambda, \ell \in \mathbb{N}$, $\mathbf{x} \in \mathfrak{A}$, and $\mathbf{y} \in \mathfrak{P}$, letting $ct_{\mathbf{x}} \longleftarrow Encrypt(pp, SK, \mathbf{x}, M)$ and $sk_{\mathbf{y}} \longleftarrow KeyGen(pp, SK, \mathbf{y})$, we have

$$\begin{aligned} M &\longleftarrow Decrypt\left(pp, ct_{\mathbf{x}}, sk_{\mathbf{y}}\right), \text{if } P(\mathbf{x}, \mathbf{y}) = 1; \\ \perp &\longleftarrow Decrypt\left(pp, ct_{\mathbf{x}}, sk_{\mathbf{y}}\right), \text{if } P(\mathbf{x}, \mathbf{y}) = 0, \end{aligned} \quad (4)$$

where $(pp, SK) \longleftarrow Setup(1^{\lambda}, 1^{\ell})$.

In this paper, we construct a private PE scheme supporting inner product predicate function defined over $\mathbb{Z}_q^{\ell}$, where q is a large prime. That is,

(i) ℓ denotes the dimension of the vector space.

(ii) $\mathfrak{A} = \mathfrak{P} = \mathbb{Z}_p^{\ell}$.

(iii) For all $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_q^{\ell}$, P: $(\mathbf{x}, \mathbf{y}) = 1$ if $\langle \mathbf{x}, \mathbf{y} \rangle = 0$.

Such encryption schemes are called "private PE for inner product" (private IPE), and $\mathfrak{A} = \mathbb{Z}_q^{\ell}$ and $\mathfrak{P} = \mathbb{Z}_q^{\ell}$ are called attribute vector space and predicate vector space, respectively.

3.2. Security Definitions. In private PE, there exist three types of adversary that want to retrieve the information of message, ciphertext attribute, and key attribute from ciphertext and secret key. Therefore, we model three security requirements of private PE, payload-hiding, attribute-hiding, and predicate-hiding securities, to model the attacks from these adversaries.

The payload-hiding security [7] for predicate function $P\colon \mathfrak{A} \times \mathfrak{P} \longrightarrow \{0,1\}$ is defined as an interactive game between a challenger $\mathscr{C}$ and an adversary $\mathscr{A}$. In payload-hiding models, a ciphertext reveals nothing about the encrypted message, and thus in some literature it is defined as IND-CPA security.

3.2.1. Payload-Hiding Game

(i) *Setup.* The challenger $\mathscr{C}$ runs $Setup(1^{\lambda}, 1^{\ell})$ to generate a secret key SK and the system parameter pp. Then, it sends the system parameter pp to the adversary $\mathscr{A}$ and keeps the secret key SK secretly.

(ii) *Query Phase 1.* $\mathscr{A}$ can query polynomially many times of the oracles described as follows:

  (i) KeyGen Oracle: when $\mathscr{A}$ issues a query with $\mathbf{y} \in \mathfrak{P}$, $\mathscr{C}$ returns a secret key $sk_{\mathbf{y}} \longleftarrow KeyGen(pp, SK, \mathbf{y})$.

  (ii) Encrypt Oracle: when $\mathscr{A}$ issues a query with $\mathbf{x} \in \mathfrak{A}$ and a message M, $\mathscr{C}$ returns a ciphertext $ct_{\mathbf{x}} \longleftarrow Encrypt(pp, SK, \mathbf{x}, M)$.

(iii) *Challenge.* The adversary $\mathscr{A}$ submits $\mathbf{x}^* \in \mathfrak{A}$ such that $P(\mathbf{x}^*, \mathbf{y}) = 0$ for all $\mathbf{y} \in \mathfrak{P}$, which has been queried to KeyGen Oracle in Query Phase 1, and two massages M_0, M_1 with the same length to the challenger $\mathscr{C}$. Then $\mathscr{C}$ randomly chooses $b \in \{0,1\}$ and returns a challenge ciphertext $c_{\mathbf{x}^*} \longleftarrow Encrypt(pp, SK, \mathbf{x}^*, M_b)$.

(iv) *Query Phase 2.* This phase is the same as Query Phase 1, except that $\mathscr{A}$ is only allowed to make a query to KeyGen Oracle with $\mathbf{y} \in \mathfrak{P}$ such that $P(\mathbf{x}^*, \mathbf{y}) = 0$.

(v) *Guess.* The adversary $\mathscr{A}$ outputs a bit b' and wins the game if $b' = b$.

The advantage of an adversary for winning the payload-hiding game is defined as

$$\mathrm{Adv}_{\mathscr{A}}^{PH}(1^{\lambda}) = \left|\Pr[b' = b] - \frac{1}{2}\right|. \tag{5}$$

Definition 5 (payload-hiding for private predicate encryption). We say that private PE is payload-hiding if there is no probabilistic polynomial-time adversary $\mathscr{A}$ winning the above payload-hiding game with a nonnegligible advantage.

Next, we define the "attribute-hiding" security for private PE, which can be also extended from the attribute-hiding definition for conventional PE [7]. Attribute-hiding security models that there is no adversary can obtain any information of the ciphertext attribute $\mathbf{x}$ from the ciphertext. We then define attribute-hiding via a security game between a challenger $\mathscr{C}$ and an adversary $\mathscr{A}$.

3.2.2. Attribute-Hiding Game

(i) Setup, Query Phase 1, Query Phase 2, and Guess are the same as those in the payload-hiding game.

(ii) *Challenge.* The adversary $\mathscr{A}$ submits two ciphertext attributes $\mathbf{x}^{(0)}, \mathbf{x}^{(1)} \in \mathfrak{A}$ such that $P(\mathbf{x}^{(0)}, \mathbf{y}) = P(\mathbf{x}^{(1)}, \mathbf{y})$ for all $\mathbf{y} \in \mathfrak{P}$, which has been queried to KeyGen Oracle in Query Phase 1, and a massage M with the same length to the challenger $\mathscr{C}$. Then, $\mathscr{C}$ randomly chooses $b \in \{0,1\}$ and returns a challenge ciphertext $c_{\mathbf{x}^*} \longleftarrow Encrypt(pp, SK, \mathbf{x}^{(b)}, M)$.

The advantage of an adversary for winning the attribute-hiding game is defined as

$$\mathrm{Adv}_{\mathscr{A}}^{AH}(1^{\lambda}) = \left|\Pr[b' = b] - \frac{1}{2}\right|. \tag{6}$$

Definition 6 (attribute-hiding for private predicate encryption). We say that private PE is attribute-hiding, if there is no probabilistic polynomial-time adversary $\mathscr{A}$ winning the above attribute-hiding game with a nonnegligible advantage.

There is another weaker notion, called "weak attribute-hiding" [25]. The weak attribute-hiding game is the same as the above attribute-hiding game, except the following:

(i) The adversary sends $(\mathbf{x}^{(0)}, M_0)$, $(\mathbf{x}^{(1)}, M_1)$ to invoke the Challenge phase.

(ii) The restriction on $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}$ is modified to "$P(\mathbf{x}^{(0)}, \mathbf{y}) = P(\mathbf{x}^{(1)}, \mathbf{y}) = 0$ for all $\mathbf{y} \in \mathfrak{P}$ which has been queried to KeyGen Oracle in Query Phase 1."

Furthermore, we define the "predicate-hiding" for private PE scheme via the following game, which models the notion that a secret key $sk_{\mathbf{y}}$ reveals nothing about the key attribute $\mathbf{y}$.

3.2.3. Predicate-Hiding Game

(i) Setup, Query Phase 1, Query Phase 2, and Guess are the same as those in the payload-hiding game.

(ii) *Challenge.* The adversary $\mathscr{A}$ submits two key attributes $\mathbf{y}^{(0)}, \mathbf{y}^{(1)} \in \mathfrak{P}$ to the challenger $\mathscr{C}$, such that $P(\mathbf{x}, \mathbf{y}^{(0)}) = P(\mathbf{x}, \mathbf{y}^{(1)}) = 0$ for all $\mathbf{x} \in \mathfrak{A}$ which has been queried to Encrypt Oracle in Query Phase 1. Then, $\mathscr{C}$ randomly chooses $b \in \{0,1\}$ and returns a challenge secret key $sk_{\mathbf{y}^{(b)}} \longleftarrow KeyGen(pp, SK, \mathbf{y}^{(b)})$.

The advantage of an adversary for winning the predicate-hiding game is defined as

$$\mathrm{Adv}_{\mathscr{A}}^{PP}(1^{\lambda}) = \left|\Pr[b' = b] - \frac{1}{2}\right|. \tag{7}$$

Definition 7 (predicate-hiding for private predicate encryption). We say that private PE achieves predicate-hiding if there is no probabilistic polynomial-time adversary $\mathscr{A}$ winning the above predicate-hiding game with nonnegligible advantage.

4. A Private IPE from Key-Homomorphic PRF

In the following, we describe how to obtain a private IPE from a key-homomorphic PRF. In our construction, we require that $\mathscr{R} = \mathbb{Z}_q$, for some prime q. Additionally, we assume that the decryptor knows the value of predicate vector $\mathbf{y}$ of his/her secret key $sk_{\mathbf{y}}$.

(i) $Setup(1^{\lambda}, 1^{\ell})$. Suppose that the message space is $\{0,1\}^m$ for some positive integer $m = poly(\lambda)$. Given the security parameters (λ, ℓ), where $\lambda, \ell \in \mathbb{N}$, the algorithm outputs the system parameter pp and the secret key SK as follows:

    (i) Choose a prime $q = poly(\lambda)$.
    (ii) Choose a key-homomorphic PRF $F\colon \mathscr{R} \times \mathscr{X} \longrightarrow \mathscr{R}$, where $\mathscr{X}$ is the domain and $(\mathscr{R}, +, \cdot)$ is a ring.
    (iii) Choose $\mathbf{a} = (a_1, \ldots, a_{\ell}) \in \mathscr{R}^{\ell}$.
    (iv) Choose a pseudorandom generator $G\colon \mathscr{R} \longrightarrow \{0,1\}^m$.
    (v) Choose $h \overset{\$}{\longleftarrow} \mathscr{X}$.
    (vi) Output $pp = (F, G)$ and the secret key $SK = (\mathbf{a}, h)$.

    Note that the descriptions of F and G are implicitly included in the system parameter pp.

(ii) $Encrypt(pp, SK, \mathbf{x}, M)$. Given the system parameter pp, a secret key SK, an attribute vector $\mathbf{x} = (x_1, \ldots, x_{\ell}) \in \mathscr{R}^{\ell}$, and a message M, the algorithm runs the following steps:

    (i) Choose random $\delta, \sigma \overset{\$}{\longleftarrow} \mathscr{R}$.
    (ii) $c_i = F(\sum_{j=1}^{\delta} (x_i) + a_i, h) + \sigma$ for $i = 1, \ldots, \ell$.
    (iii) $c_0 = M \oplus G(\sigma)$.
    (iv) Output ciphertext $ct_{\mathbf{x}} = (c_0, \ldots, c_{\ell}) \in \{0,1\}^m \times \mathscr{R}^{\ell}$.

(iii) $KeyGen(pp, SK, \mathbf{y})$. Given the system parameter pp, a secret key SK, and a predicate vector $\mathbf{y} = (y_1, \ldots, y_{\ell}) \in \mathscr{R}^{\ell}$, the algorithm computes the following steps:

    (i) $sk_{\mathbf{y}} = (\sum_{i=1}^{\ell} \sum_{j=1}^{y_i} F(a_i, h)) = F(\langle \mathbf{a}, \mathbf{y} \rangle, h)$.
    (ii) Output $sk_{\mathbf{y}}$.

(iv) $Decrypt(pp, ct_{\mathbf{x}}, sk_{\mathbf{y}})$. Given the system parameter pp, a ciphertext $ct_{\mathbf{x}}$, and a secret key $sk_{\mathbf{y}}$, the algorithm computes the following steps:

    (i) $ct' = \sum_{i=1}^{\ell} (y_i \cdot c_i) - sk_{\mathbf{y}}$.
    (ii) Compute $\sigma = ct' \cdot (\sum_{i=1}^{\ell} y_i)^{-1}$.
    (iii) Compute $M = c_0 \oplus G(\sigma)$.

Correctness. Let $ct_{\mathbf{x}}$ and $sk_{\mathbf{y}}$ be as above. Then,

$$\begin{aligned} ct' &= \sum_{i=1}^{\ell} (y_i \cdot c_i) - sk_{\mathbf{y}} \\ &= \sum_{i=1}^{\ell} \left(y_i \cdot \left(F\left(\sum_{j=1}^{\delta} x_i + a_i, h \right) + \sigma \right)\right) - sk_{\mathbf{y}} \\ &= F\left(\sum_{j=1}^{\delta} \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{a}, \mathbf{y} \rangle, h \right) + \sum_{i=1}^{\ell} y_i \sigma - F(\langle \mathbf{a}, \mathbf{y} \rangle, h). \end{aligned} \tag{8}$$

If $\langle \mathbf{x}, \mathbf{y} \rangle = 0$, we have

$$ct' = F(\langle \mathbf{a}, \mathbf{y} \rangle, h) + \sum_{i=1}^{\ell} y_i \sigma - F(\langle \mathbf{a}, \mathbf{y} \rangle, h) = \sum_{i=1}^{\ell} y_i \sigma. \tag{9}$$

Then, we can compute $\sigma = ct' \cdot (\sum_{i=1}^{\ell} y_i)^{-1}$, and the plaintext can be decrypted by

$$c_0 \oplus G(\sigma) = M \oplus G(\sigma) \oplus G(\sigma) = M. \tag{10}$$

Our scheme accommodates approximate homomorphism [26], as long as the error term is bounded.

5. Security Proofs

5.1. Payload-Hiding Security. We prove the payload-hiding security of our scheme using the sequence-of-game approach [27]. Let $(c_0, c_1, \ldots, c_{\ell})$ be the challenge ciphertext given to the adversary in the payload-hiding game. Besides, let R_0 be a random element in $\{0,1\}^m$ and let $R_1, \ldots, R_{\ell}$ be random elements in $\mathscr{R}$. We define the following hybrid games differing in what challenge ciphertext is sent to the adversary:

(i) $Game_0$. The challenge ciphertext is $(c_0, c_1, \ldots, c_{\ell})$. It is identical to the original payload-hiding game defined in Section 3.2.
(ii) $Game_i, 1 \leq i \leq \ell$. The challenge ciphertext is $(c_0, R_1, \ldots, R_i, c_{i+1}, \ldots, c_{\ell})$.
(iii) $Game_{\ell+1}$. The challenge ciphertext is $(R_0, R_1, \ldots, R_{\ell})$.

We remark that the challenge ciphertext in $Game_{\ell+1}$ leaks no information about the encrypted message, since it is composed of $\ell + 1$ random elements, whereas the challenge ciphertext in $Game_0$ is well formed. Therefore, the advantage of the adversary in the last game is 0. We then prove the indistinguishability between the adjacent games in the following lemmas.

Lemma 1. *If the underlying PRF F is secure, then $Game_{k-1}$ is indistinguishable from $Game_k$, for $k = 1, \ldots, \ell$.*

Proof. Suppose that there is an adversary $\mathscr{A}$ that is able to distinguish $Game_{k-1}$ from $Game_k$ with a nonnegligible advantage. Then we can build a challenger $\mathscr{C}_1$ to distinguish the experiment $EXP(0)$ from the experiment $EXP(1)$

shown in Section 2. After invoking the experiment $EXP(b)$ and receiving the description of the PRF F, the challenger $\mathscr{C}_1$ simulates a hybrid game for an adversary $\mathscr{A}$ as follows:

Setup. The challenger first randomly chooses $a_1, \ldots, a_{k-1}, a_{k+1}, \ldots, a_\ell$ from $\mathscr{R}$ and h from $\mathscr{X}$ and a pseudorandom generator G and then sends $pp = (F, G)$ to the adversary. Next, the challenger makes a Challenge query with h to the underlying experiment and obtains f as the response. The value of f will be used in the later simulation for KeyGen and Encryption Oracle.

Query Phase 1. In this phase, the adversary is allowed to make polynomially many queries to the following oracles.

(i) KeyGen Oracle: taking as inputs a vector $\mathbf{y} = (y_1, \ldots, y_\ell) \in \mathscr{R}^\ell$, the challenger computes

$$\begin{aligned} sk_{\mathbf{y}} &= \sum_{j=1}^{y_1} F(a_1, h) + \cdots + \sum_{j=1}^{y_{k-1}} F(a_{k-1}, h) \\ &\quad + \sum_{j=1}^{y_k} f + \sum_{j=1}^{y_{k+1}} F(a_{k+1}, h) + \cdots + \sum_{j=1}^{y_\ell} F(a_\ell, h) \\ &= \sum_{\substack{i=1 \\ i \neq k}}^{\ell} y_i F(a_i, h) + y_k f, \end{aligned} \quad (11)$$

and returns $sk_{\mathbf{y}}$ to the adversary. By implicitly setting a_j to the chosen key of the underlying experiment, it is easy to verify that $sk_{\mathbf{y}}$ is a valid secret key for $\mathbf{y}$.

(ii) Encryption Oracle: taking as inputs a vector $\mathbf{x} = (x_1, \ldots, x_\ell) \in \mathscr{R}^\ell$ and a message M, the challenger performs as follows:

(1) Randomly choose δ, σ from $\mathscr{R}$.

(2) Compute $c_k = F(\sum_{j=1}^{\delta}\ (x_k), h) + f + \sigma = F(\delta x_k, h) + f + \sigma$.

(3) For $i = 1, \ldots, k-1, k+1, \ldots, \ell$, compute c_i the same as in the *Encrypt* algorithm since the challenger knows $a_1, \ldots, a_{k-1}, a_{k+1}, \ldots, a_\ell, h$.

(4) Compute $c_0 = M \oplus G(\sigma)$.

(5) Return $ct_{\mathbf{x}} = (c_0, c_1, \ldots, c_\ell)$.

Challenge. The adversary submits two messages M_0, M_1 with the same length and a vector $\mathbf{x}^* = (x_1^*, \ldots, x_\ell^*)$, such that $\langle \mathbf{x}^*, \mathbf{y} \rangle \neq 0$ for all $\mathbf{y}$ queried to KeyGen Oracle. After receiving $\mathbf{x}^*, M_0, M_1$, the challenger randomly chooses $\beta \xleftarrow{\$}$ and then can compute the challenge ciphertext ct^* as follows:

(1) Randomly choose δ, σ from $\mathscr{R}$.

(2) For $i = 1, \ldots, \ell$,

(i) if $i < k$, choose a random element $R_i \xleftarrow{\$}$ and set $c_i = R_i$.

(ii) if $i = k$, compute

$$c_k = F\left(\sum_{j=1}^{\delta} (x_k^*, h)\right) + f + \sigma = F(\delta x_k^*, h) + f + \sigma. \quad (12)$$

(iii) if $i > k$, compute c_i the same way as that in the scheme since the challenger knows $a_{k+1}, \ldots, a_\ell$ and h.

(3) Compute $c_0 = M_\beta \oplus G(\sigma)$.

(4) Return $ct^* = (c_0, c_1, \ldots, c_\ell)$.

Query Phase 2. It is the same as Query Phase 1 except that the adversary is not allowed to make a query to KeyGen Oracle with $\mathbf{y}$ such that $\langle \mathbf{x}^*, \mathbf{y} \rangle = 0$.

Guess. The adversary outputs a bit β'. Then the challenger outputs 1 if $\beta' = \beta$ and 0 otherwise. Before analyzing the advantages of the challenger in breaking the underlying PRF, we first discuss that the outputs of the oracles are well formed, no matter which experiment the challenger interacts with. Let S_i be the event where the adversary makes a right guess in $Game_i$. First, if the challenger is actually interacting with the experiment $EXP(0)$, then f is a random element in $\mathscr{R}$. In this case, the answer to a KeyGen Oracle,

$$sk_{\mathbf{y}} = \sum_{\substack{i=1 \\ i \neq k}}^{\ell} y_i F(a_i, h) + y_k f, \quad (13)$$

is an element of $\mathscr{R}$ and the answer to an Encryption query $(c_0, c_1, \ldots, c_\ell)$ is a vector in $\{0,1\}^m \times \mathscr{R}^\ell$, and

$$\begin{aligned} \sum_{i=1}^{\ell} (y_i \cdot c_i) - sk_{\mathbf{y}} &= \sum_{\substack{i=1 \\ i \neq k}}^{\ell} y_i (F(\delta x_i + a_i, h) + \sigma) \\ &\quad + y_k (F(\delta x_k, h) + f + \sigma) \\ &\quad - \left(\sum_{\substack{i=1 \\ i \neq k}}^{\ell} y_i F(a_i, h) + y_k f \right) \\ &= \sum_{i=1}^{\ell} y_i F(\delta x_i, h) + \sum_{i=1}^{\ell} \sigma \\ &= F\left(\delta \sum_{i=1}^{\ell} x_i y_i, h \right) + \sum_{i=1}^{\ell} \sigma \\ &= \sum_{i=1}^{\ell} \sigma \\ \Longleftrightarrow \langle \mathbf{x}, \mathbf{y} \rangle &= 0. \end{aligned} \quad (14)$$

Therefore, the answers to KeyGen and Encryption queries are well formed.

Next, we analyze the advantage of $\mathscr{C}_1$ in breaking the underlying PRF. First, if the challenger is interacting with the experiment $EXP(0)$, then f is a random element in $\mathscr{R}$. Thus, $c_1, \ldots, c_k$ in the challenge ciphertext are random elements, and thus we are in $Game_k$. Thus, the probability that the challenger outputs 1 is

$$\Pr[S_k] = \Pr[\mathscr{C}_1 \text{ outputs } 1] = \Pr[\beta' = \beta] = \Pr[W_0]. \quad (15)$$

Second, if the challenger is interacting with the experiment $EXP(1)$, then f is the output of the PRF with input h. By implicitly setting the encryption key component a_k as the chosen key of the underlying experiment, we have $f = F(a_k, h)$, and thus the challenger answers the KeyGen and Encryption queries correctly. As for the challenge ciphertext, we have that

$$c_k = F\left(\sum_{j=1}^{\delta}(x_k^*), h\right) + f + \sigma = F(\delta x_k^*, h) + F(a_k, h) + \sigma = F(\delta x_k^* + a_k, h) + \sigma, \quad (16)$$

is a valid ciphertext component. Since $c_1, \ldots, c_{k-1}$ are random elements from $\mathscr{R}$, we are in $Game_{k-1}$. In this case, the probability that the challenger outputs 1 is

$$\Pr[S_{k-1}] = \Pr[\mathscr{C}_1 \text{ outputs } 1] = \Pr[\beta' = \beta] = \Pr[W_1]. \quad (17)$$

Finally, combining the above two cases, we have that

$$|\Pr[S_{k-1}] - \Pr[S_k]| = |\Pr[W_1] - \Pr[W_0]| = \mathrm{Adv}_{\mathscr{C}_1}^{\mathrm{PRF}}(1^\lambda), \quad (18)$$

and hence, $Game_{k-1}$ is indistinguishable from $Game_k$, if the underlying pseudorandom function is secure, for $k = 1, \ldots, \ell$. □

Lemma 2. *If the underlying PRG G is secure, then $Game_\ell$ is indistinguishable from $Game_{\ell+1}$.*

Proof. Given the description of the PRG G and a challenge $\psi \in \{0,1\}^m$, the challenger $\mathscr{C}_2$ simulates the following hybrid game for an adversary $\mathscr{A}$:

Setup. The challenger first chooses a key-homomorphic pseudorandom function $F\colon \mathscr{R} \times \mathscr{X} \longrightarrow \mathscr{R}$, $a_1, \ldots, a_\ell$ from $\mathscr{R}$ and h from $\mathscr{X}$ and then sends (F, G) to the adversary.

Query Phase 1. The challenger is able to answer the KeyGen (Encryption, resp.) queries by following the *KeyGen* (*Encrypt*, resp.) algorithms to generate the secret keys $sk_{\mathbf{y}}$ (ciphertexts $ct_{\mathbf{x}}$, resp.), since the challenger knows the secret key $SK = (a_1, \ldots, a_\ell, h)$.

Challenge. The adversary submits two messages M_0, M_1 with the same length and a vector $\mathbf{x}^*$, such that $\langle \mathbf{x}^*, \mathbf{y}\rangle \neq 0$ for all $\mathbf{y}$ queried to KeyGen Oracle. After receiving $\mathbf{x}^*, M_0, M_1$, the challenger randomly chooses $\beta \xleftarrow{\$} \{0,1\}$ and then can compute the challenge ciphertext ct^* as follows:

(1) Randomly choose $R_1, \ldots, R_\ell \xleftarrow{\$} \mathscr{R}$.
(2) For $i = 1, \ldots, \ell$, set $c_i = R_i$.
(3) Compute $c_0 = M_\beta \oplus \psi$.
(4) Return the challenge ciphertext $ct^* = (c_0, c_1, \ldots, c_\ell)$.

Query Phase 2. It is the same as Query Phase 1 except that the adversary is not allowed to make a query to KeyGen Oracle with $\mathbf{y}$ such that $\langle \mathbf{x}^*, \mathbf{y}\rangle = 0$.

Guess. The adversary outputs a bit β'. Then, the challenger outputs 1 if $\beta' = \beta$. Let S_i be the event where the adversary makes a right guess in $Game_i$. If the term $\psi = G(\sigma)$ is generated from the PRG G for some σ, then we are in $Game_\ell$, and we have

$$\Pr[S_\ell] = \Pr[\mathscr{C}_2(\psi = G(\sigma)) = 1]. \quad (19)$$

If ψ is randomly chosen from $\{0,1\}^m$, then we are in $Game_{\ell+1}$, and we have

$$\Pr[S_{\ell+1}] = \Pr\left[\mathscr{C}_2\left(\psi \xleftarrow{\$} \{0,1\}^m\right) = 1\right]. \quad (20)$$

Finally, we have that

$$\begin{aligned}|\Pr[S_\ell] - \Pr[S_{\ell+1}]| &= \left|\Pr[\mathscr{C}_2(\psi = G(\sigma)) = 1] - \Pr\left[\mathscr{C}_2\left(\psi \xleftarrow{\$} \{0,1\}^m\right) = 1\right]\right| \\ &= \mathrm{Adv}_{\mathscr{C}_2}^{\mathrm{PRG}}(1^\lambda)\end{aligned} \quad (21)$$

is negligible. □

Theorem 1. *The proposed private IPE scheme achieves payload-hiding, if the underlying pseudorandom function is key-homomorphic and secure and the pseudorandom generator is secure.*

Proof. By combining Lemmas 1 and 2, we have

$$\begin{aligned}|\Pr[S_0] - \Pr[S_{\ell+1}]| &= \left|\sum_{i=1}^{\ell}(\Pr[S_{i-1}] - \Pr[S_i]) + (\Pr[S_\ell] - \Pr[S_{\ell+1}])\right| \\ &\leq |\Pr[S_0] - \Pr[S_1]| + \cdots + \Pr[S_{\ell-1}] - \Pr[S_\ell] + |\Pr[S_\ell] - \Pr[S_{\ell+1}]| \\ &= \underbrace{\mathrm{Adv}_{\mathscr{C}_1}^{\mathrm{PRF}}(1^\lambda) + \cdots + \mathrm{Adv}_{\mathscr{C}_1}^{\mathrm{PRF}}(1^\lambda)}_{\ell} + \mathrm{Adv}_{\mathscr{C}_2}^{\mathrm{PRG}}(1^\lambda) \\ &= \ell \cdot \mathrm{Adv}_{\mathscr{C}_1}^{\mathrm{PRF}}(1^\lambda) + \mathrm{Adv}_{\mathscr{C}_2}^{\mathrm{PRG}}(1^\lambda).\end{aligned} \quad (22)$$

Note that $\Pr[S_0] = \mathrm{Adv}_{\mathscr{A}}^{PH}(1^\lambda)$ since $Game_0$ is the payload-hiding game, and $\Pr[S_{\ell+1}] = 0$ since ct^* leaks no information about the encrypted message in $Game_{\ell+1}$. Therefore, for any PPT adversary $\mathscr{A}$, there exist algorithms $\mathscr{C}_1, \mathscr{C}_2$ such that

$$\mathrm{Adv}_{\mathscr{A}}^{PH}(1^\lambda) = \Pr[S_0] = \left|\Pr[S_0] - \Pr[S_{\ell+1}]\right| \leq \ell \cdot \mathrm{Adv}_{\mathscr{C}_1}^{\mathrm{PRF}}(1^\lambda) + \mathrm{Adv}_{\mathscr{C}_2}^{\mathrm{PRG}}(1^\lambda). \tag{23}$$

is negligible. □

5.2. Attribute-Hiding Security. We then prove that our scheme achieves attribute-hiding. The proof is similar to the proof for payload-hiding security, and hence we will omit some content to avoid the unnecessary redundancy. Let $(c_0, c_1, \ldots, c_\ell)$ be the challenge ciphertext given to the adversary in the attribute-hiding game. Besides, let $R_1, \ldots, R_\ell$ be random elements in $\mathscr{R}$ and let R_0 be a random element in $\{0,1\}^m$. We define the following hybrid games differing in what challenge ciphertext is sent to the adversary:

(i) $Game_0$. The challenge ciphertext is $(c_0, c_1, \ldots, c_\ell)$. It is identical to the original attribute-hiding game defined in Section 3.2.

(ii) $Game_i, 1 \leq i \leq \ell$. The challenge ciphertext is $(c_0, R_1, \ldots, R_i, c_{i+1}, \ldots, c_\ell)$.

(iii) $Game_{\ell+1}$. The challenge ciphertext is $(R_0, R_1, \ldots, R_\ell)$.

In the last game, the challenge ciphertext is composed of $\ell + 1$ random elements, and hence the adversary obtains no information about the attribute vector from the challenge ciphertext. We then prove that the adjacent games are indistinguishable in the following lemmas.

Lemma 3. *If the underlying PRF F is secure, then $Game_{k-1}$ is indistinguishable from $Game_k$, for $k = 1, \ldots, \ell$.*

Proof. Suppose that there is an adversary $\mathscr{A}$ that is able to distinguish $Game_{k-1}$ from $Game_k$ with a nonnegligible advantage. Then we can build a challenger $\mathscr{C}_3$ to distinguish the experiment $EXP(0)$ from the experiment $EXP(1)$ shown in Section 2. After invoking the experiment $EXP(b)$ and receiving the description of the PRF F, the challenger $\mathscr{C}_1$ simulates a hybrid game for an adversary $\mathscr{A}$ as follows.

For Setup, Query Phase 1, Query Phase 2, and Guess, the challenger performs the same as in the proof of Lemma 1.

For Challenge phase, after receiving $\mathbf{x}^{(0)} = (x_1^{(0)}, \ldots, x_\ell^{(0)}), \mathbf{x}^{(1)} = (x_1^{(1)}, \ldots, x_\ell^{(1)})$, and M from the adversary, where

$$\langle \mathbf{x}^{(0)}, \mathbf{y}\rangle = 0 = \langle \mathbf{x}^{(1)}, \mathbf{y}\rangle \text{ or } \langle \mathbf{x}^{(0)}, \mathbf{y}\rangle \neq 0 \neq \langle \mathbf{x}^{(1)}, \mathbf{y}\rangle, \tag{24}$$

for all $\mathbf{y}$ queried to KeyGen Oracle in Query Phase 1, the challenger performs as follows:

(1) Randomly choose $\beta \xleftarrow{\$} \{0,1\}$.

(2) Randomly choose δ, σ from $\mathscr{R}$.

(3) For $i = 1, \ldots, \ell$,

(i) if $i < k$, choose a random element $R_i \xleftarrow{\$} \mathscr{R}$ and set $c_i = R_i$.

(ii) if $i = k$, compute the following.

$$c_k = F\left(\sum_{j=1}^{\delta}\left(x_k^{(\beta)}\right), h\right) + f + \sigma = F\left(\delta x_k^{(\beta)}, h\right) + f + \sigma. \tag{25}$$

(iii) if $i > k$, compute c_i the same way as that in the scheme since the challenger knows $a_{k+1}, \ldots, a_\ell$ and h.

(4) Compute $c_0 = M \oplus G(\sigma)$.

(5) Return $ct^* = (c_0, c_1, \ldots, c_\ell)$.

The analysis of the correctness of the simulation is similar to that in the proof of Lemma 1. Let S_i be the event where the adversary makes a right guess in $Game_i$. If f from the PRF game is a random element in $\mathscr{R}$, then we are in $Game_k$; otherwise, we are in $Game_{k-1}$. Therefore, we have

$$\left|\Pr[S_{k-1}] - \Pr[S_k]\right| = \left|\Pr[W_1] - \Pr[W_0]\right| = \mathrm{Adv}_{\mathscr{C}_3}^{\mathrm{PRF}}(1^\lambda). \tag{26}$$

That is, $Game_k$ is indistinguishable from $Game_{k-1}$, if the underlying pseudorandom function is secure, for $k = 1, \ldots, \ell$. □

Lemma 4. *If the underlying PRG G is secure, then $Game_\ell$ is indistinguishable from $Game_{\ell+1}$.*

Proof. The proof of this lemma is similar to the proof of Lemma 2, with the only difference being that the challenger received two vectors $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}$ with a message M; in Lemma 2, the challenger received two messages M_0, M_1 with a vector $\mathbf{x}^*$ from the adversary.

Given the description of the PRG G and a challenge $\psi \in \{0,1\}^m$, the challenger $\mathscr{C}_4$ simulates the following hybrid game for an adversary $\mathscr{A}$.

For Setup, Query Phase 1, Query Phase 2, and Guess, the challenger performs the same as in the proof of Lemma 1.

For Challenge phase, after receiving $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}$, and M from the adversary, where

$$\langle \mathbf{x}^{(0)}, \mathbf{y}\rangle = 0 = \langle \mathbf{x}^{(1)}, \mathbf{y}\rangle \text{ or } \langle \mathbf{x}^{(0)}, \mathbf{y}\rangle \neq 0 \neq \langle \mathbf{x}^{(1)}, \mathbf{y}\rangle, \tag{27}$$

for all $\mathbf{y}$ queried to KeyGen Oracle in Query Phase 1, the challenger performs as follows:

(1) Randomly choose $R_1, \ldots, R_\ell \xleftarrow{\$} \mathscr{R}$.

(2) For $i = 1, \ldots, \ell$, set $c_i = R_i$.

(3) Compute $c_0 = M \oplus \psi$.

(4) Return the challenge ciphertext $ct^* = (c_0, c_1, \ldots, c_\ell)$.

The analysis of the correctness of the simulation is similar to that in the proof of Lemma 3. Let S_i be the event where the adversary makes a right guess in $Game_i$. If ψ from the PRG game is a random element in $\{0,1\}^m$, then we are in $Game_{\ell+1}$; otherwise, we are in $Game_\ell$. Therefore, we have that

$$\begin{aligned}\left|\Pr[S_\ell] - \Pr[S_{\ell+1}]\right| &= \left|\Pr[\mathscr{C}_4(\psi = G(\sigma)) = 1] - \Pr\left[\mathscr{C}_4\left(\psi \xleftarrow{\$} \{0,1\}^m\right) = 1\right]\right| \\ &= \mathrm{Adv}^{\mathrm{PRG}}_{\mathscr{C}_4}\left(1^\lambda\right),\end{aligned} \tag{28}$$

is negligible. That is, $Game_\ell$ is indistinguishable from $Game_{\ell+1}$, if the underlying pseudorandom generator is secure. □

Theorem 2. *The proposed private IPE scheme achieves attribute-hiding, if the underlying pseudorandom function is key-homomorphic and secure and the pseudorandom generator is secure.*

Proof. By combining Lemma 3 and Lemma 4, we have

$$\begin{aligned}\left|\Pr[S_0] - \Pr[S_{\ell+1}]\right| &= \left|\sum_{i=1}^{\ell}\left(\Pr[S_{i-1}] - \Pr[S_i]\right) + \left(\Pr[S_\ell] - \Pr[S_{\ell+1}]\right)\right| \\ &\leq \left|\Pr[S_0] - \Pr[S_1]\right| + \cdots + \Pr[S_{\ell-1}] - \Pr[S_\ell] + \left|\Pr[S_\ell] - \Pr[S_{\ell+1}]\right| \\ &= \underbrace{\mathrm{Adv}^{\mathrm{PRF}}_{\mathscr{C}_3}\left(1^\lambda\right) + \cdots + \mathrm{Adv}^{\mathrm{PRF}}_{\mathscr{C}_3}\left(1^\lambda\right)}_{\ell} + \mathrm{Adv}^{\mathrm{PRG}}_{\mathscr{C}_4}\left(1^\lambda\right) \\ &= \ell \cdot \mathrm{Adv}^{\mathrm{PRF}}_{\mathscr{C}_3}\left(1^\lambda\right) + \mathrm{Adv}^{\mathrm{PRG}}_{\mathscr{C}_4}\left(1^\lambda\right).\end{aligned} \tag{29}$$

Note that $\Pr[S_0] = \mathrm{Adv}^{AH}_{\mathscr{A}}(1^\lambda)$ since $Game_0$ is the attribute-hiding game, and $\Pr[S_{\ell+1}] = 0$ since ct^* leaks no information about the encrypted message in $Game_{\ell+1}$. Therefore, for any PPT adversary $\mathscr{A}$, there exist algorithms $\mathscr{C}_3, \mathscr{C}_4$ such that

$$\mathrm{Adv}^{AH}_{\mathscr{A}}\left(1^\lambda\right) = \Pr[S_0] = \left|\Pr[S_0] - \Pr[S_{\ell+1}]\right| \leq \ell \cdot \mathrm{Adv}^{\mathrm{PRF}}_{\mathscr{C}_3}\left(1^\lambda\right) + \mathrm{Adv}^{\mathrm{PRG}}_{\mathscr{C}_4}\left(1^\lambda\right) \tag{30}$$

is negligible. □

5.3. Predicate-Hiding Security. We first give the intuition for the proof. Let $[y_1, y_2, \ldots, y_\ell]$ denote the challenge secret key generated using the vector $(y_1, y_2, \ldots, y_\ell)$. Besides, let $\mathbf{y}^{(0)} = (y_1^{(0)}, y_2^{(0)}, \ldots, y_\ell^{(0)}), \mathbf{y}^{(1)} = (y_1^{(1)}, y_2^{(1)}, \ldots, y_\ell^{(1)})$ be the two vectors sent from the adversary in the Challenge phase. To prove the indistinguishability between the cases $[y_1^{(0)}, y_2^{(0)}, \ldots, y_\ell^{(0)}]$ and $[y_1^{(1)}, y_2^{(1)}, \ldots, y_\ell^{(1)}]$ given to the adversary, we define a sequence of games below and show the indistinguishability of any two adjacent games. Each game differs in the challenge secret key given to the adversary. Let $y_1', y_2', \ldots, y_\ell'$ be random elements from $\mathscr{R}$.

$Game_{0,i}$: $[y_1', y_2', \ldots, y_{k-1}', y_k^{(0)}, \ldots, y_\ell^{(0)}]$ is given $(k = 1, \ldots, \ell)$

$Game_{1,i}$: $[y_1', y_2', \ldots, y_{k-1}', y_k^{(1)}, \ldots, y_\ell^{(1)}]$ is given $(k = 1, \ldots, \ell)$

Note that $Game_{0,\ell}$ and $Game_{1,\ell}$ are identical, and $Game_{0,0}$ and $Game_{1,0}$ are the games where $[y_1^{(0)}, \ldots, y_\ell^{(0)}]$ and $[y_1^{(1)}, \ldots, y_\ell^{(1)}]$ are given to the adversary, respectively. We then give the following lemma to prove that

$$Game_{0,0} \approx Game_{0,1} \approx \cdots \approx Game_{0,\ell} \approx Game_{1,\ell} \approx \cdots Game_{1,1} \approx \cdots \approx Game_{1,0}. \tag{31}$$

Lemma 5. *If the underlying PRF F is secure, then $Game_{0,k-1}$ and $Game_{0,k}$ are indistinguishable, for $k = 1, \ldots, \ell$.*

Proof. Suppose that there is an adversary $\mathscr{A}$ which is able to distinguish $Game_{k-1}$ from $Game_k$ with a nonnegligible advantage. Then we can build a challenger $\mathscr{C}$ to distinguish the experiment $EXP(0)$ from $EXP(1)$ shown in Section 2. After invoking the experiment $EXP(b)$ and receiving the description of the PRF F, the challenger $\mathscr{C}$ simulates a hybrid game for an adversary $\mathscr{A}$ as follows.

Table 1: Comparison with other related private IPE schemes [17–19]. Here, the length of ciphertext attribute and key attribute is n. $|\mathbb{G}|$ and m represent size of an element of $|\mathbb{G}|$ and message, respectively. MSK, SK, CT, Qun. Res., GSD, C3DH, and 3FCOBGA stand for master secret key, secret key for some key attribute, ciphertext for some ciphertext attribute, quantum-resistant, general subgroup decision, composite 3-party (decisional) Diffie-Hellman, and 3-factor-based composite-order bilinear groups assumption, respectively.

	SSW09 [18]	YKNS12 [19]	KT13 [17]	Ours
Security	Selective	Selective	Adaptive	Adaptive
Order of $\mathbb{G}$	Composite	Composite	Prime	—
Assumption	A variant of GSD, C3DH, DLIN	3FCOBGA	DLIN	PRF
MSK size	$(4n+4)\|\mathbb{G}\|$	$(4n+4)\|\mathbb{G}\|$	$5n\|\mathbb{G}\|$	$n\log_2 q$
SK size	$(2n+2)\|\mathbb{G}\|$	$(2n+2)\|\mathbb{G}\|$	$6n\|\mathbb{G}\|$	$\log_2 q$
CT size	$(2n+2)\|\mathbb{G}\|$	$(2n+2)\|\mathbb{G}\|$	$6n\|\mathbb{G}\|$	$m+n\log_2 q$
Qun. Res.	No	No	No	Yes†

†If the underlying PRF is resistant to quantum attacks, then our proposed scheme is resistant to quantum attacks.

For Setup, Query Phase 1, Query Phase 2, and Guess, the challenger performs the same as in the proof of Lemma 1.

For Challenge phase, after receiving $\mathbf{y}^{(0)} = (y_1^{(0)},\ldots,y_\ell^{(0)}), \mathbf{y}^{(1)} = (y_1^{(1)},\ldots,y_\ell^{(1)})$ from the adversary, where

$$\langle \mathbf{x},\mathbf{y}^{(0)}\rangle = 0 = \langle \mathbf{x},\mathbf{y}^{(1)}\rangle \text{ or } \langle \mathbf{x},\mathbf{y}^{(0)}\rangle \neq 0 \neq \langle \mathbf{x},\mathbf{y}^{(1)}\rangle, \quad (32)$$

for all $\mathbf{x}$ queried to Encrypt Oracle in Query Phase 1, the challenger performs as follows.

(1) Randomly choose $y_1', y_2', \ldots, y_{k-1}'$ from $\mathscr{R}$.

(2) Compute

$$sk^* = \sum_{j=1}^{y_1'} F(a_1,h) + \ldots + \sum_{j=1}^{y_{k-1}'} F(a_{k-1},h) + \sum_{j=1}^{y_k^{(0)}} f + \sum_{j=1}^{y_{k+1}^{(0)}} F(a_{k+1},h) + \cdots + \sum_{j=1}^{y_\ell^{(0)}} F(a_\ell,h). \quad (33)$$

(3) Return sk^*.

If the challenger is interacting with the experiment $EXP(1)$, then f is the output of the PRF with input h. By implicitly setting the encryption key component a_k as the chosen key of the underlying experiment, we have $f = F(a_k,h)$, and thus we have

$$\begin{aligned} sk^* &= y_1' F(a_1,h) + \cdots + y_{k-1}' F(a_{k-1},h) + y_k^{(0)} F(a_k,h) \\ &\quad + y_{k+1}^{(0)} F(a_{k+1},h) + \cdots + y_\ell^{(0)} F(a_\ell,h) \\ &= \left[y_1',\ldots,y_{k-1}',y_k^{(0)},\ldots,y_\ell^{(0)}\right], \end{aligned} \quad (34)$$

and thus we are in $Game_{k-1}$. Otherwise, f is a random element in $\mathscr{R}$; then we can rewrite $f = F(a_k,h) + \widetilde{R}$ for some random element $\widetilde{R} \in \mathscr{R}$. Besides, there must exist an element $\widetilde{y}$ such that $\widetilde{R} = (y_k^{(0)})^{-1}\widetilde{y}F(a_k,h)$. By implicitly setting $y_k' = y_k^{(0)} + \widetilde{y}$, we have

$$\begin{aligned} \sum_{i=1}^{y_k^{(0)}} f &= y_k^{(0)} f = y_k^{(0)}\left(F(a_k,h) + \widetilde{R}\right) = y_k^{(0)} F(a_k,h) + y_k^{(0)}\left(y_k^{(0)}\right)^{-1}\widetilde{y}F(a_k,h) \\ &= y_k^{(0)} F(a_k,h) + \widetilde{y}F(a_k,h) = \left(y_k^{(0)} + \widetilde{y}\right)F(a_k,h) = y_k' F(a_k,h). \end{aligned} \quad (35)$$

Since f is a random element in $\mathscr{R}$, y'_k is also a random element in $\mathscr{R}$. That means $sk^* = [y'_1, \ldots, y'_k, y^{(0)}_{k+1}, \ldots, y^{(0)}_{\ell}]$, and thus we are in $Game_k$. Let S_i be the event where the adversary makes a right guess in $Game_i$. Therefore, we have

$$\left|\Pr\left[S_{k-1}\right] - \Pr\left[S_k\right]\right| = \left|\Pr\left[W_1\right] - \Pr\left[W_0\right]\right| = \mathrm{Adv}^{\mathrm{PRF}}_{\mathscr{C}}\left(1^{\lambda}\right). \tag{36}$$

That is, $Game_k$ is indistinguishable from $Game_{k-1}$, if the underlying PRF is secure, for $k = 1, \ldots, \ell$. □

Theorem 3. *The proposed private IPE scheme achieves predicate-hiding, if the underlying pseudorandom function is key-homomorphic and secure and the pseudorandom generator is secure.*

Proof. The proof for the indistinguishability between $Game_{1,k-1}$ and $Game_{1,k}$ is the same as that for the indistinguishability between $Game_{0,k-1}$ and $Game_{0,k}$, due to the symmetry of the game sequence. This completes the proof of the predicate-hiding. □

6. Comparison and Analysis

To the best of our knowledge, although existing private IPE schemes [17–19] can resist payload-hiding, attribute-hiding, and predicate-hiding security, these schemes cannot resist quantum attacks because their security is based on discrete logarithm assumption. In this section, we compare our scheme with the existing private IPE schemes in terms of security properties and the size of master secret key, secret key, and ciphertext, as shown in Table 1.

The results show that our construction has higher security and efficiency in terms of secret key size because the size is not related to attribute length. In particular, the security of [18, 19] is only selective security; meanwhile that in [17] and our construction is adaptive security, making it more resistant to real attacks. In secret key size, our construction is of constant size, while the secret key sizes of [17–19] are linearly related to the key attribute length. In terms of ciphertext size, the encryption algorithm in schemes [17–19] only encrypts ciphertext predicate, while our proposed construction further encrypts message; therefore, the ciphertext size of our scheme is $m + n \log_2 q$, where m is the length of message. Finally, [17–19] are not resistant to quantum attacks, while our construction is resistant to quantum attacks if the underlying PRF is resistant to quantum attacks.

7. Conclusions and Future Works

With the development of cloud computing, the privacy of uploaded data needs to be concerned and protected. Private IPE is well suited to cloud computing scenario because it provides encryption for access control. In this paper, we propose a generic private IPE construction that achieves payload-hiding, attribute-hiding, and predicate-hiding security by utilizing specific key-homomorphic PRF. For future works, because the current construction requires that the key space and output space of the underlying key-homomorphic PRF be $\mathbb{Z}_q$, how to provide construction with less restriction is an open problem that remains to be solved.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the Ministry of Science and Technology, Taiwan (ROC), under Project nos. MOST 108-2218-E-004-001-, MOST 108-2218-E-004-002-MY2, MOST 109-2218-E-011-007-, and by Taiwan Information Security Center at National Sun Yat-sen University (TWISC@NSYSU).

References

[1] L. Zhou, X. Li, K.-H. Yeh, C. Su, and W. Chiu, "Lightweight IoT-based authentication scheme in cloud computing circumstance," *Future Generation Computer*, vol. 91, pp. 244–251, 2019.

[2] S. Kumari, P. Chaudhary, C.-M. Chen, and M. K. Khan, "Questioning key compromise attack on Ostad-Sharif et al.'s authentication and session key generation scheme for healthcare applications," *IEEE Access*, vol. 7, pp. 39717–39720, 2019.

[3] P. Wang, C.-M. Chen, S. Kumari, M. Shojafar, R. Tafazolli, and Y. N. Liu, "HDMA: hybrid D2D message authentication scheme for 5G-enabled VANETs," *IEEE Access*, vol. 2020, 2020.

[4] C.-M. Chen, B. Xiang, K.-H. Wang, K.-H. Yeh, and T.-Y. Wu, "A robust mutual authentication with a key agreement scheme for session initiation protocol," *Applied Sciences*, vol. 8, no. 10, p. 1789, 2018.

[5] J. Li, S. Wang, Y. Li et al., "An efficient attribute-based encryption scheme with policy update and file update in cloud computing," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6500–6509, 2019.

[6] M. Zhang, Yu Chen, and S. E. P. P. F. M. Jiajun Huang., "A searchable encryption scheme supporting privacy-preserving fuzzy multikeyword in cloud systems," 2020.

[7] J. Katz, S. Amit, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *EUROCRYPT 2008. LNCS*, N. Smart, Ed., vol. 4965, pp. 146–162, Springer, Berlin, Heidelberg, 2008.

[8] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," *CCS*, vol. 417–426, 2008.

[9] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in *TCC 2007. LNCS*, S. P. Vadhan, Ed., vol. 4932, pp. 535–554, Springer, Berlin, Heidelberg, 2007.

[10] V. Iovino and G. Persiano, "Hidden-vector encryption with groups of prime order," in *Pairing 2008. LNCS*, S. D Galbraith and K. G. Paterson, Eds., vol. 5209, pp. 75–88, Springer, Berlin, Heidelberg, 2008.

[11] M. Naor and B. Pinkas, “Oblivious transfer and polynomial evaluation,” *STOC*, vol. 1999, pp. 245–254, 1999.

[12] A. Ge, R. Zhang, C. Chen, C. Ma, and Z. Zhang, “Threshold ciphertext policy attribute-based encryption with constant size ciphertexts,” in *ACISP 2012. LNCS*, W. Susilo, Y. Mu, and J. Seberry, Eds., vol. 7372, pp. 336–349, pringer, Berlin, Heidelberg, 2012.

[13] S. Agrawal, D. M. Freeman, and V. Vaikuntanathan, “Functional encryption for inner product predicates from learning with errors,” in *ASIACRYPT 2011. LNCS*, D. H. Lee and X. Wang, Eds., vol. 7073, pp. 21–40, Springer, Berlin, Heidelberg, 2011.

[14] T. Okamoto and K. Takashima, “Hierarchical predicate encryption for inner-products,” in *ASIACRYPT 2009. LNCS*, M. Matsui, Ed., vol. 5912, pp. 214–231, Springer, Berlin, Heidelberg, 2009.

[15] T. Okamoto and K. Takashima, “Adaptively attribute-hiding (hierarchical) inner product encryption,” in *EUROCRYPT 2012. LNCS*, D. Pointcheval and T. Johansson, Eds., vol. 7237, pp. 591–608, Springer, Berlin, Heidelberg, 2012.

[16] K. Xagawa, “Improved (hierarchical) inner-product encryption from lattices,” in *PKC 2013. LNCS*, K. Kurosawa and G. Hanaoka, Eds., vol. 7778, pp. 235–252, Springer, Berlin, Heidelberg, 2013.

[17] Y. Kawai and K. Takashima, “Predicate- and attribute-hiding inner product encryption in a public key setting,” in *Pairing 2013. LNCS*, Z. Cao and F. Zhang, Eds., vol. 8365, pp. 113–130, Springer, Berlin, Heidelberg, 2013.

[18] E. Shen, E. Shi, and B. Waters, “Predicate privacy in encryption systems,” in *TCC 2009. LNCS*, O. Reingold, Ed., vol. 5444, pp. 457–473, Springer, Berlin, Heidelberg, 2009.

[19] M. Yoshino, N. Kunihiro, K. Naganuma, and H. Sato, “Symmetric inner-product predicate encryption based on three groups,” in *ProvSec 2012. LNCS*, T. Takagi, G. Wang, Z. Qin, S. Jiang, and Y. Yu, Eds., vol. 7496, pp. 215–234, Springer, Berlin, Heidelberg, 2012.

[20] P. W. Shor, “Algorithms for quantum computation: discrete logarithms and factoring,” *FOCS*, vol. 1994, pp. 124–134, 1994.

[21] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” *SIAM Review*, vol. 41, no. 2, pp. 303–332, 1999.

[22] N. Alamati, H. Montgomery, and S. Patranabis, “Ring key-homomorphic weak PRFs and applications,” 2020.

[23] O. Goldreich, S. Goldwasser, and S. Micali, “How to construct random functions,” *Journal of the ACM*, vol. 33, no. 4, pp. 792–807, 1986.

[24] D. Boneh, K. Lewi, H. Montgomery, and A. Raghunathan, “Key homomorphic PRFs and their applications,” in *CRYPTO 2013. LNCS*, R. Canetti and J. A. Garay, Eds., vol. 8042, pp. 410–428, Springer, Berlin, Heidelberg, 2013.

[25] J. Chen and J. Gong, “ABE with tag made easy,” in *ASIACRYPT 2017. LNCS*, T. Takagi and T. Peyrin, Eds., vol. 10625, pp. 35–65, Springer, Berlin, Heidelberg, 2017.

[26] D. Boneh, S. Eskandarian, S. Kim, and M. Shih, “Improving speed and security in updatable encryption schemes,” 2020.

[27] V. Shoup, “Sequences of games: a tool for taming complexity in security proofs,” 2004.