Practical Predicate Encryption for Inner Product*

Yi-Fan Tseng, Zi-Yuan Liu† and Raylin Tso
Department of Computer Science, National Chengchi University, Taipei, Taiwan
{*yftseng, zyliu, raylin*}*@cs.nccu.edu.tw*

Keywords: Predicate Encryption, Inner Product Encryption, Constant-size Private Key, Efficient Decryption, Constant Pairing Computations

Abstract: Inner product encryption is a powerful cryptographic primitive, where a private key and a ciphertext are both associated with a predicate vector and an attribute vector, respectively. A successful decryption requires the inner product of the predicate vector and the attribute vector to be zero. Most of the existing inner product encryption schemes suffer either long private key or heavy decryption cost. In this manuscript, an efficient inner product encryption is proposed. The length for a private key is only an element in $\mathbb{G}$ and an element in $\mathbb{Z}_p$. Besides, only one pairing computation is needed for decryption. Moreover, both formal security proof and implementation result are demonstrated in this manuscript. To the best of our knowledge, our scheme is the most efficient one in terms of the private key length and the number of pairings computation for decryption.

1 INTRODUCTION

Traditional public key encryption provides only coarse-grained access control. That is, given a ciphertext encrypted under a public key, only the owner of the corresponding private key can obtain the plaintext. However, in many applications, such as distributed file systems and cloud services, more complex access policies may be necessary. Compared with traditional public key encryption, predicate encryption (Boneh and Waters, 2007; Katz et al., 2008) can provide fine-grained access control over encrypted data. Such encryption is suitable for various applications, for instance, searching over encrypted data. In a predicate encryption scheme, the ciphertext for message M is associated with an attribute x, and the private key is associated with a predicate f. A successful decryption requires that $f(x) = 1$.

Katz et al. (2008) first considers the predicate for the computation of inner product over $\mathbb{Z}_N$, where N is a composite number. They also gave an instance for inner product predicate, called inner product encryption (IPE). In an IPE scheme, the ciphertext associated with an attribute vector $\mathbf{x}$ can be decrypted by the private key associated with a predicate vector $\mathbf{y}$, if and only if $\langle \mathbf{x}, \mathbf{y} \rangle = 0$ (Here $\langle \mathbf{x}, \mathbf{y} \rangle$ denotes the standard inner product operation for vectors $\mathbf{x}, \mathbf{y}$). Due to its flexibleness, lots of works on IPE scheme have been proposed, such as pairing-based IPE schemes (Okamoto and Takashima, 2009, 2015; Kurosawa and Phong, 2017; Chen et al., 2018; Zhang et al., 2019) and lattice-based IPE schemes (Agrawal et al., 2011; K. Xagawa, 2013; Li et al., 2017; Wang et al., 2018).

In additional to its usefulness on fine-grained access control, IPE scheme can be used to construct various cryptographic primitives or be convert to more complex primitives, e.g., identity-based encryption (IBE) (Shamir, 1985; Boneh and Franklin, 2001; Boneh and Boyen, 2004), hidden vector encryption (Boneh and Waters, 2007; Lee, 2017), subset predicate encryption (Katz et al., 2018; Chatterjee and Mukherjee, 2019). We refer the readers to Katz et al. (2008, 2018) for details.

Although many IPE schemes have been proposed, these schemes suffer from either large private key sizes or heavy computation costs, as described below:

- *Pairing-based IPE schemes*: existing pairing-based IPE schemes are generally computationally inefficient because of the large number of pairings (linear to vector lengths) used during decryption. In addition, the private key length of most schemes is also linear to vector lengths, so it is

*An extended abstract of this paper appears at SECRYPT 2020. This is the full version.

†Corresponding author.

not practical enough.

- *Lattice-based IPE schemes*: though lattice-based IPE schemes are believed to be quantum-resistant, nearly all of them suffer from either large key size, or small message space.

All the problems mentioned above will make an IPE scheme impractical and brings us to the following open question:

Can we optimize the length of the private key and reduce the cost of decryption, and further make them constant in relation to vector lengths?

1.1 Contributions

In this manuscript, we give a positive answer to the above question by proposing an effective inner product encryption scheme. More preciously, in the proposed scheme, the length of a private key is only an element in $\mathbb{G}$ and an element in $\mathbb{Z}_p$, i.e., independent of the length of the predicate vector. Besides, the decryption is efficient since only one pairing is necessary (also independent of the length of the predicate vector). We also provide rigours proof to show that our proposed scheme is co-selective IND-CPA secure under modified decisional Diffie-Hellman assumption. Furthermore, Table 1 and Table 3 show the comparison with other state-of-the-art schemes, illustrating that our proposed scheme is not only secure, but also very practical.

1.2 Related Works

1.2.1 Pairing-based IPE Schemes

The first IPE scheme was introduced by Katz et al. (2008), which allows evaluating predicates over $\mathbb{Z}_N$ using inner product, where N is a composite number. After this pioneering work, many studies have been proposed. Okamoto and Takashima (2009) proposed the first hierarchical predicate encryption (or delegateable predicate encryption) for inner product predicates, which allows a user with functionality that can delegate more restrictive functionality to another user. Attrapadung and Libert (2010) constructed an IPE scheme, which solves the inefficiency of the previous scheme. More preciously, as long as the description of the ciphertext attribute vector is not included in the ciphertext, the ciphertext overhead of the scheme is reduced to $O(1)$. Lewko et al. (2010), by carefully combining dual system encryption (Waters, 2009) and dual pairing vector spaces (Okamoto and Takashima, 2009), obtained the first fully secure IPE scheme and hierarchical predicate encryption under n-extended decisional Diffie-Hellman assumption. However, the security of all previous studies based on non-standard assumptions. In order to solve this issue, Park (2011) provided the first IPE scheme under the standard assumptions (i.e., decisional bilinear Diffie-Hellman (DBDH) and decisional linear (DLIN) assumptions). Okamoto and Takashima (2011) then introduced two non-zero inner-product encryption (NIPE) schemes that support constant-size ciphertexts and constant-size secret key respectively, which are adaptively secure under the DLIN assumption in the standard model. Okamoto and Takashima (2012a) proposed the first IPE scheme that is fully secure and fully attribute-hiding, and Okamoto and Takashima (2012b) further proposed the first unbounded IPE scheme that is also fully secure and fully attribute-hiding in the standard model under DLIN assumption. Kawai and Takashima (2014) introduced a new notion, called IPE with ciphertext conversion, which takes into account the security of predicate hiding. Zhenlin and Wei (2015) introduced another concept, called multiparty cloud computation IPE with multiplicative homomorphic property, which enables IPE to support multiparty cloud computation. Kim et al. (2016) proposed a new efficient IPE scheme which only required n exponentiation and three pairing computations for decryption. Huang et al. (2016) proposed the first enabled/disabled IPE, which supports timed-release services and data self-destruction. Ramanna (2016) constructed two IPE schemes using tag-based quasi-adaptive non-interactive zero-knowledge. The former has the property of constant-size ciphertext, while the latter has the same property as the former and has the property of attribute hiding. Zhang et al. (2019) proposed a new IPE scheme based on double encryption system, which is proven to be adaptive security under weak attribute hiding model.

As mentioned below, although a lot of work has been proposed, the private key length of most schemes is linearly dependent on the vector length, or requires many pairing operations, making these schemes impractical. Thus, how to construct a more practical scheme still an important issue.

1.2.2 Lattice-based IPE Schemes

On the other hand, to fend off the attack of quantum computers in the future, Agrawal et al. (2011) proposed the first IPE scheme based on lattice hard assumption (i.e., learning with error assumption, which is believed to be able to withstand quantum attacks) by twisting an identity-based encryption (Agrawal et al., 2010). K. Xagawa (2013), inspired

by Agrawal et al.'s work, proposed an improved lattice-based IPE scheme that reduced the size of public parameters and ciphertext. Li et al. (2017) proposed a lattice-based IPE scheme that further reduced the size of public parameters and ciphertext. In contract to K. Xagawa (2013), the work reduces the size by a factor of $\log n$, where n is security parameter. Recently, Wang et al. (2018) proposed the first compact IPE scheme by employing IPE scheme (K. Xagawa, 2013), fully homomorphic encryption (Gentry et al., 2013), and vector encoding schemes (Apon et al., 2017).

Although these constructions are thought to be able to withstand quantum computer attacks, they are based on the LWE assumption, resulting in key lengths that are still too large to be practical.

1.3 Organization

The remainder of this paper is organized as follows. We start by some preliminaries on bilinear maps, complexity assumptions, and the definition of inner product encryption in Section 2. In Section 3, we propose our inner product encryption scheme and show its correcness. In Section 4, we demonstrate security proofs using modified decisional Diffie-Hellman problem. In Section 5 and 6, we give a comparison with other state-of-the-art schemes and show the implementation result. Finally, we conclude this paper in section 7.

2 PRELIMINARIES

In this section we give the necessary preliminaries, such as notations, complex assumptions, and the definition for IPE scheme.

2.1 Notations

Given a set S, "choose an element x randomly from the set S" will be denoted as "$x \xleftarrow{\$} S$". We use $x \leftarrow A$ to denote "x is the output of the algorithm A". The bold lowercase latter, e.g., $\mathbf{s}$, is used to denote a vector. For a vector $\mathbf{s}$, $\mathbf{s}_i$ denotes the i-th entry of vector $\mathbf{s}$. Given two vectors $\mathbf{x}, \mathbf{y}$, we denote the inner product of these two vectors as $\langle \mathbf{x}, \mathbf{y} \rangle$. The set of positive integer and integer are represented by $\mathbb{N}$ and $\mathbb{Z}$, respectively. For a prime p, $\mathbb{Z}_p$ denotes the set of integers module p.

2.2 Bilinear Maps

Let $\mathbb{G}$ be an additive cyclic group and $\mathbb{G}_T$ be a multiplicative cyclic group, where the order of $\mathbb{G}$ and $\mathbb{G}_T$ is a large prime p (i.e., $|\mathbb{G}| = |\mathbb{G}_T| = p$). Besides, let P be a generator of $\mathbb{G}$. A bilinear map (pairing) $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a mapping with the following properties.

- **Bilinearity**. For $a, b \in \mathbb{Z}_p$, $e(aP, bP) = e(P, P)^{ab}$.
- **Non-Degeneracy**. $\exists P \in \mathbb{G}$, such that $e(P, P) \neq 1_{\mathbb{G}_T}$.
- **Computability**. The mapping e is efficiently computable.

2.3 Complexity Assumption

In this work, we take advantage of the generalized decisional Diffie-Hellman exponent (GDDHE) problem due to Boneh et al. (2005). The GDDHE problem is a generic framework to create new complexity assumptions. We first give an overview of the GDDHE problem. Let

- p be a prime;
- s, n be two positive integers;
- $P, Q \in \mathbb{F}_p[X_1, \ldots, X_n]^s$ be two s-tuple of n-variate polynomials over $\mathbb{F}_p$;
- f be a n-variate polynomial in $\mathbb{F}_p[X_1, \ldots, X_n]$.

Note that Q, Q_T are two ordered sets with multivariate polynomials, and thus we denote $Q = (q_1, q_2, \ldots, q_s)$ and $R = (r_1, r_2, \ldots, r_s)$. As stated in Boneh et al. (2005), we require $p_1 = q_1 = 1$ to be two constant polynomials. Consider a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ with the generator P of $\mathbb{G}$ and $g_T = e(P, P) \in \mathbb{G}_T$. For a vector $(x_1, x_2, \ldots, x_n) \in \mathbb{F}_p^n$, we define

$$\begin{aligned} & Q(x_1, x_2, \ldots, x_n)P \\ = \; & (q_1(x_1, x_2, \ldots, x_n)P, \ldots, q_s(x_1, x_2, \ldots, x_n)P) \in \mathbb{G}^s, \end{aligned}$$

and

$$\begin{aligned} & g_T^{R(x_1, x_2, \ldots, x_n)} \\ = \; & (g_T^{r_1(x_1, x_2, \ldots, x_n)}, \ldots, g_T^{r_s(x_1, x_2, \ldots, x_n)}) \in \mathbb{G}_T^s. \end{aligned}$$

By "f depends on (Q, R)" we mean that there are $s^2 + s$ constants $\{a_{i,j}\}_{i,j=1}^s$ and $\{b_k\}_{k=1}^s$ such that

$$f = \sum_{i,j=1}^{s} a_{i,j} q_i q_j + \sum_{k=1}^{s} b_k r_k.$$

We say that f is independent of (Q, R) if f is not depend on (Q, R).

Definition 1 (The (Q,R,f)-GDDHE Problem)**.** *Given* $(Q(x_1,\ldots,x_n)P, g_T^{R(x_1,\ldots,x_n)}, Z) \in \mathbb{G}^s \times \mathbb{G}_T^s \times \mathbb{G}_T$*, decide if* $Z \stackrel{?}{=} g_T^{f(x_1,\ldots,x_n)}$*. For an algorithm* $\mathcal{A}$*, the advantage of* $\mathcal{A}$ *in solving the* (Q,R,f)*-GDDHE problem is defined as*

$$\begin{aligned} & \mathbf{Adv}^{(Q,R,f)\text{-GDDHE}}(\mathcal{A}) \\ = & \Big|\mathcal{A}(Q(x_1,\ldots,x_n)P, g_T^{R(x_1,\ldots,x_n)}, g_T^{f(x_1,\ldots,x_n)}) \\ - & \mathcal{A}(Q(x_1,\ldots,x_n)P, g_T^{R(x_1,\ldots,x_n)}, Z \stackrel{\$}{\leftarrow} \mathbb{G}_T)\Big|. \end{aligned}$$

In Boneh et al.'s paper, they have proposed that the (Q,R,f)-GDDHE problem is hard if f is independent of (Q,R). They also show a large class of hard problems can be fit into the framework of the GDDHE problem, e.g., the DDH problem over $\mathbb{G}_T$.

Definition 2 (The Decisional Diffie-Hellman Problem over $\mathbb{G}_T$ ($\text{DDH}_{\mathbb{G}_T}$ problem))**.** *Let* $g_T = e(P,P)$ *be a generator of* $\mathbb{G}_T$*. Given* $(P, g_T, A = g_T^a, B = g_T^b, C) \in \mathbb{G} \times \mathbb{G}_T^4$*, where* $a,b \stackrel{\$}{\leftarrow} \mathbb{Z}_p$*, decide whether* $C = g_T^{ab}$ *or an random element from* $\mathbb{G}_T$*.*

By setting $Q = (1), R = (1,a,b), f = ab$, the DDH problem over $\mathbb{G}_T$ is equivalent to the (Q,R,f)-GDDHE problem. Observe that there exist no constants such that the linear combination of $1,a,b$ equals to ab, f is therefore independent of (Q,R). From Boneh et al.'s result, we can say that there is no algorithm to solve the $\text{DDH}_{\mathbb{G}_T}$ problem with non-negligible advantage. We refer the readers to Boneh et al. (2005) for more details.

Next, we give a modified version of the $\text{DDH}_{\mathbb{G}_T}$ problem which will be used in the security proof.

Definition 3 (The Modified Decisional Diffie-Hellman Problem over $\mathbb{G}_T$ ($\text{M-DDH}_{\mathbb{G}_T}$ problem))**.** *Let* $g_T = e(P,P)$ *be a generator of* $\mathbb{G}_T$*. Given* $(P, A' = aP, g_T, A = g_T^a, B = g_T^b, C) \in \mathbb{G}^2 \times \mathbb{G}_T^4$*, where* $a,b \stackrel{\$}{\leftarrow} \mathbb{Z}_p$*, decide whether* $C = g_T^{ab}$ *or an random element from* $\mathbb{G}_T$*.*

Compared with the $\text{DDH}_{\mathbb{G}_T}$ problem, the instance of the $\text{M-DDH}_{\mathbb{G}_T}$ problem contains an additional element $A' = aP$. One can observe that the $\text{M-DDH}_{\mathbb{G}_T}$ problem is equivalent to the (Q,R,f)-GDDHE problem with

$$Q = (1,a), R = (1,a,b), f = ab.$$

We can see that there exist no constants such that the linear combination of the monomials $(1 \cdot a), 1, a, b$ equals to the polynomial ab. Therefore, from Boneh et al.'s result, we conclude that the $\text{M-DDH}_{\mathbb{G}_T}$ is hard.

Besides, define the advantage for an algorithm $\mathcal{D}$ in solving the $\text{M-DDH}_{\mathbb{G}_T}$ problem as

$$\begin{aligned} & \mathbf{Adv}^{\text{M-DDH}_{\mathbb{G}_T}}(\mathcal{D}) \\ = & \big|\Pr[\mathcal{D}(P,A',g_T,A,B,C = g_T^{ab}) = 1] \\ - & \Pr[\mathcal{D}(P,A',g_T,A,B,C \stackrel{\$}{\leftarrow} \mathbb{G}_T) = 1]\big|. \end{aligned}$$

Definition 4 (The Modified Decisional Diffie-Hellman Assumption over $\mathbb{G}_T$ ($\text{M-DDH}_{\mathbb{G}_T}$ assumption))**.** *We say that the* $\text{M-DDH}_{\mathbb{G}_T}$ *assumption holds if there is no algorithm* $\mathcal{D}$ *solving the* $\text{M-DDH}_{\mathbb{G}_T}$ *problem with a non-negligible advantage.*

2.4 Definition of Inner Product Encryption

An inner product encryption scheme consists of four algorithms: **Setup**, **KeyGen**, **Encrypt**, **Decrypt**. The details of the algorithms are shown below.

Setup$(1^\lambda, 1^\ell)$. Taken as input the security parameters $(1^\lambda, 1^\ell)$, where $\lambda, \ell \in \mathbb{N}$, the algorithm outputs the system parameter $\texttt{params}$ and the master secret key $\texttt{msk}$. Note that the description of the attribute vector space $\mathfrak{A}$ and the predicate vector space $\mathfrak{P}$ will be implicitly included in $\texttt{params}$. Besides, we require that the inner product operation over $\mathfrak{A}$ and $\mathfrak{P}$ should be well-defined.

Encrypt$(\texttt{params}, \mathbf{x}, M)$. Given the system parameter $\texttt{params}$, an attribute vector $\mathbf{x} \in \mathfrak{A}$, and a message M, the algorithm outputs a ciphertext $\texttt{C}_\mathbf{x}$ for the attribute vector $\mathbf{x}$.

KeyGen$(\texttt{params}, \texttt{msk}, \mathbf{y})$. Given the system parameter $\texttt{params}$, a predicate vector $\mathbf{y} \in \mathfrak{P}$, the algorithm outputs the private key $\texttt{K}_\mathbf{y}$ for the predicate vector $\mathbf{y}$.

Decrypt$(\texttt{params}, \texttt{C}_\mathbf{x}, \texttt{K}_\mathbf{y})$. Given the system parameter $\texttt{params}$, a ciphertext $\texttt{C}_\mathbf{x}$, and the private key $\texttt{K}_\mathbf{y}$, the algorithm output a message M or a error symbol $\perp$.

The correctness is defined as follows. For all $\lambda, \ell \in \mathbb{N}$, let $\texttt{C}_\mathbf{x} \leftarrow$ **Encrypt**$(\texttt{params}, \mathbf{x} \in \mathfrak{A}, M)$ and let $\texttt{K}_\mathbf{y} \leftarrow$ **KeyGen**$(\texttt{params}, \texttt{msk}, \mathbf{y} \in \mathfrak{P})$, we have

$$\begin{aligned} M &\leftarrow \textbf{Decrypt}(\texttt{params}, \texttt{C}_\mathbf{x}, \texttt{K}_\mathbf{y}) \quad \text{if } \langle \mathbf{x}, \mathbf{y} \rangle = 0; \\ \perp &\leftarrow \textbf{Decrypt}(\texttt{params}, \texttt{C}_\mathbf{x}, \texttt{K}_\mathbf{y}) \quad \text{if } \langle \mathbf{x}, \mathbf{y} \rangle \neq 0, \end{aligned}$$

where $(\texttt{params}, \texttt{msk}) \leftarrow$ **Setup**$(1^\lambda, 1^\ell)$.

2.5 Security Model

Here, we first introduce the IND-CPA security for inner product encryption.

The IND-CPA game of an inner product encryption for attribute vector space $\mathfrak{A}$ and predicate vector space $\mathfrak{P}$ is defined as an interactive game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Setup. The challenger $\mathcal{C}$ runs **Setup**$(1^\lambda, 1^\ell)$ and sends the system parameter $\mathtt{params}$ to the adversary $\mathcal{A}$.

Query Phase 1. The challenger answers polynomially many private key queries for $\mathbf{y} \in \mathfrak{P}$ for the adversary $\mathcal{A}$ by returning $\mathtt{K}_\mathbf{y} \leftarrow$ **KeyGen**$(\mathtt{params}, \mathtt{msk}, \mathbf{y})$.

Challenge. The adversary $\mathcal{A}$ submits an attribute vector $\mathbf{x}^* \in \mathfrak{A}$ such that $\langle \mathbf{x}^*, \mathbf{y} \rangle \neq 0$ for all $\mathbf{y}$ which has been queried in **Query Phase 1**, and two massages M_0, M_1 with the same length to the challenger $\mathcal{C}$. Then $\mathcal{C}$ randomly chooses $\beta \in \{0,1\}$ and returns a challenge ciphertext $\mathtt{C}_{\mathbf{x}^*} \leftarrow$ **Encrypt**$(\mathtt{params}, \mathbf{x}^*, M_\beta)$.

Query Phase 2. This phase is the same as **Query Phase 1**, except that the adversary is not allowed to make a query with $\mathbf{y} \in \mathfrak{P}$ such that $\langle \mathbf{x}^*, \mathbf{y} \rangle \neq 0$.

Guess. The adversary $\mathcal{A}$ outputs a bit β' and wins the game if $\beta' = \beta$. The advantage of an adversary for winning the IND-CPA game is defined as

$$\mathbf{Adv}^{\mathsf{IND\text{-}CPA}}(\mathcal{A}) = \left| \Pr[\beta' = \beta] - \frac{1}{2} \right|.$$

Definition 5 (IND-CPA Security for Inner Product Encryption)**.** *We say that an inner product encryption is IND-CPA secure if there is no probabilistic polynomial-time adversary $\mathcal{A}$ wins the IND-CPA game with a non-negligible advantage.*

In some literature (Katz et al., 2008; Park, 2011), the security notions for an inner product encryption is defined with the notions "payload hiding" and "attribute hiding". Informally, payload hiding (resp. attribute hiding) is defined to argue that a ciphertext leaks no information about the encrypted message (resp. attribute vector). The IND-CPA security shown in this section is equivalent to payload hiding.

We then present the selective security and the co-selective security (Attrapadung and Libert, 2010; Attrapadung, 2014) for inner product encryption.

The selective IND-CPA (sIND-CPA) game is defined as the same of the IND-CPA game, except that the adversary $\mathcal{A}$ is forced to commit ahead before **Setup** phase an attribute vector $\mathbf{x}^*$, and $\mathcal{A}$ is not allowed to make private key queries with $\mathbf{y}$ such that $\langle \mathbf{x}^*, \mathbf{y} \rangle \neq 0$ in both **Query Phase 1** and **Query Phase 2**.

Definition 6 (Selective IND-CPA Security for Inner Product Encryption)**.** *An inner product encryption scheme is said to be sIND-CPA secure if no probabilistic polynomial-time adversary wins the sIND-CPA game with non-negligible advantage.*

The co-selective IND-CPA (csIND-CPA) game is defined as the same of the IND-CPA game, except that the adversary $\mathcal{A}$ is forced to commit ahead before **Setup** phase q predicate vectors $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(q)}$ for the private key queries, where q is a polynomial in the security parameter λ, and $\mathcal{A}$ is required to invoke **Challenge** phase with an attribute vector $\mathbf{x}^*$ such that $\langle \mathbf{x}^*, \mathbf{y}^{(j)} \rangle \neq 0$ for $j = 1, \ldots, q$.

Definition 7 (Co-Selective IND-CPA Security for Inner Product Encryption)**.** *An inner product encryption scheme is said to be csIND-CPA secure if no probabilistic polynomial-time adversary wins the csIND-CPA game with non-negligible advantage.*

One can think of co-selective security as the complementary notion for selective security. In the selective security game, the adversary is able to learn private key according to its previous choices, while in the co-selective security game, the adversary is able to choose its target after seeing the public parameter and learning the private keys of its choice. We note that, though selective security and co-selective security are weaker than the full security, both notions are incomparable in general by definition.

3 THE PROPOSED INNER PRODUCT ENCRYPTION SCHEME

Our IPE scheme consists of four algorithms: **Setup**, **KeyGen**, **Encrypt**, **Decrypt**. The details of the proposed scheme are demonstrated below.

Setup$(1^\lambda, 1^\ell)$. Given the security parameters $(1^\lambda, 1^\ell)$, where $\lambda, \ell \in \mathbb{N}$, the algorithm performs as follows.

1. Choose bilinear groups $\mathbb{G}, \mathbb{G}_T$ of prime order $p > 2^\lambda$. Let P and $g_T = e(P,P)$ be the generator of $\mathbb{G}$ and $\mathbb{G}_T$, respectively.

2. Set the predicate vector space and the attribute vector space to $\mathbb{Z}_p^\ell$.
3. Choose $\mathbf{s} = (s_1, s_2, \ldots, s_\ell) \xleftarrow{\$} \mathbb{Z}_p^\ell$.
4. Compute $\widehat{\mathbf{h}} = (g_T^{s_i})_{i=1}^\ell = (\widehat{h}_1, \ldots, \widehat{h}_\ell)$.
5. Output the system parameter $\mathtt{params} = (P, g_T, \widehat{\mathbf{h}})$, and the master secret key $\mathtt{msk} = \mathbf{s}$.

Encrypt$(\mathtt{params}, \mathbf{x}, M)$. Given the system parameter $\mathtt{params}$, a vector $\mathbf{x} = (x_1, x_2, \ldots, x_\ell) \in \mathbb{Z}_p^\ell$, and a message $M \in \mathbb{G}_T$, the algorithm performs as follows.

1. Choose $r, \delta \xleftarrow{\$} \mathbb{Z}_p$.
2. Compute $\mathtt{C}_0 = rP$, and $\widehat{\mathtt{C}}_0 = g_T^r$.
3. Compute $\mathtt{C}_i = \widehat{h}_i^r \cdot g_T^{\delta x_i} \cdot M$ for $i = 1$ to ℓ.
4. Output the ciphertext $\mathtt{C}_{\mathbf{x}} = (\mathtt{C}_0, \widehat{\mathtt{C}}_0, \mathtt{C}_1, \mathtt{C}_2, \ldots, \mathtt{C}_\ell)$

KeyGen$(\mathtt{params}, \mathtt{msk}, \mathbf{y})$. Given the system parameter $\mathtt{params}$, a master secret key $\mathtt{msk}$, and a vector $\mathbf{y} = (y_1, y_2, \ldots, y_\ell) \in \mathbb{Z}_p^\ell$, where $\sum_{i=1}^\ell y_i \neq 0$, the algorithm performs as follows.

1. Choose $k \xleftarrow{\$} \mathbb{Z}_p$.
2. Compute $\mathtt{K}_0 = kP$, and $\mathtt{K}_1 = \langle \mathbf{s}, \mathbf{y} \rangle + k \mod p$.
3. Output the private key $\mathtt{K}_{\mathbf{y}} = (\mathtt{K}_0, \mathtt{K}_1)$.

Decrypt$(\mathtt{params}, \mathtt{C}_{\mathbf{x}}, \mathtt{K}_{\mathbf{y}})$. Given the system parameter $\mathtt{params}$, a ciphertext $\mathtt{C}_{\mathbf{x}}$, and the private key $\mathtt{K}_{\mathbf{y}}$, where $\mathbf{y} = (y_1, y_2, \ldots, y_\ell)$ the algorithm performs as follows.

1. Compute $\mathtt{D}_0 = e(\mathtt{K}_0, \mathtt{C}_0)$.
2. Compute $\mathtt{D}_1 = \prod_{i=1}^\ell \mathtt{C}_i^{y_i}$.
3. Compute $\mathtt{D} = \dfrac{\mathtt{D}_0 \cdot \mathtt{D}_1}{\widehat{\mathtt{C}}_0^{\mathtt{K}_1}}$.
4. Compute $d = (\sum_{i=1}^\ell y_i)^{-1} \mod p$.
5. Compute $M = \mathtt{D}^d$.

3.1 Correctness

The correctness of the proposed scheme is shown as follows.

- $D_0 = e(\mathtt{K}_0, \mathtt{C}_0) = e(kP, rP) = g_T^{kr}$
- $$\begin{aligned} D_1 &= \prod_{i=1}^\ell \mathtt{C}_i^{y_i} \\ &= \prod_{i=1}^\ell (\widehat{h}_i^r \cdot g_T^{\delta x_i} \cdot M)^{y_i} \\ &= \prod_{i=1}^\ell (\widehat{h}_i^{y_i})^r \cdot (g_T^{\delta x_i y_i}) \cdot (M^{y_i}) \\ &= \prod_{i=1}^\ell ((g_T^{s_i})^{y_i})^r \prod_{i=1}^\ell (g_T^{\delta x_i y_i}) \prod_{i=1}^\ell (M^{y_i}) \\ &= g_T^{r\langle \mathbf{s}, \mathbf{y} \rangle} \cdot g_T^{\delta \langle \mathbf{x}, \mathbf{y} \rangle} \cdot M^{\sum_{i=1}^\ell y_i} \end{aligned}$$
- $\widehat{\mathtt{C}}_0^{\mathtt{K}_1} = g_T^{r\mathtt{K}_1} = g_T^{r\langle \mathbf{s}, \mathbf{y} \rangle + rk}$
- $$\begin{aligned} D &= \frac{D_0 \cdot D_1}{\widehat{\mathtt{C}}_0^{\mathtt{K}_1}} \\ &= \frac{g_T^{r\langle \mathbf{s}, \mathbf{y} \rangle} \cdot g_T^{\delta \langle \mathbf{x}, \mathbf{y} \rangle} \cdot M^{\sum_{i=1}^\ell y_i} \cdot g_T^{kr}}{g_T^{r\langle \mathbf{s}, \mathbf{y} \rangle + rk}} \\ &= g_T^{\delta \langle \mathbf{x}, \mathbf{y} \rangle} \cdot M^{\sum_{i=1}^\ell y_i} \end{aligned}$$
- We have that $D = M^{\sum_{i=1}^\ell y_i}$ iff $\langle \mathbf{x}, \mathbf{y} \rangle = 0$.
- Thus $D^d = M^{\sum_{i=1}^\ell y_i \cdot ((\sum_{i=1}^\ell y_i)^{-1} \mod p)} = M$.

4 SECURITY PROOF

We now give the security proof for the co-selective security of the proposed IPE scheme. In the following proof, we view a vector as a row vector.

Theorem 1. *The proposed scheme is csIND-CPA secure for q private key queries, where q is a polynomial in the security parameter λ, under the M-DDH$_{\mathbb{G}_T}$ assumption.*

Proof. Given $(P, A' = aP, g_T, A = g_T^a, B = g_T^b, C)$, we build an algorithm $\mathcal{C}$ using the adversary $\mathcal{A}$ to solve the M-DDH$_{\mathbb{G}_T}$ problem as follows.

Init. The adversary $\mathcal{A}$ commits q predicate vectors $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(q)}$.

Setup. $\mathcal{C}$ first finds a vector $\mathbf{u} = (u_1, u_2, \ldots, u_\ell)$ such that

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_q \end{bmatrix} \mathbf{u}^\top = \mathbf{0}_\ell^\top,$$

where $\mathbf{0}_\ell = \underbrace{(0, 0, \ldots, 0)}_{\ell}$. Such $\mathbf{u}$ exists when $q > \ell$. That is, to find a vector $\mathbf{u}$ such that $\langle \mathbf{u}, \mathbf{y}_j \rangle = 0$ for $j = 1$ to q. $\mathcal{C}$ then chooses $\mathbf{v} = (v_1, v_2, \ldots, v_\ell) \xleftarrow{\$} \mathbb{Z}_p^\ell$. Next, $\mathcal{C}$ computes $\widehat{\mathbf{h}} = (B^{u_i} \cdot g_T^{v_i})_{i=1}^\ell = (\widehat{h}_1, \ldots, \widehat{h}_\ell)$. Finally, $\mathcal{C}$ sets $\mathtt{params} = (P, g_T, \widehat{\mathbf{h}})$ and sends $\mathtt{params}$ to $\mathcal{A}$. Note

that $\mathcal{C}$ implicitly sets $\mathtt{msk} = \mathbf{s} = (s_i = u_i \cdot b + v_i)_{i=1}^{\ell}$.

Query Phase 1. After receiving $\mathbf{y}^{(i)} = (y_1^{(i)}, \ldots, y_\ell^{(i)})$ from $\mathcal{A}$, where $i \in [1,2,\ldots,q]$, $\mathcal{C}$ first chooses $k \xleftarrow{\$} \mathbb{Z}_p$, and computes $\mathtt{K}_{\mathbf{y}^{(i)}} = (\mathtt{K}_0, \mathtt{K}_1) = (kP, \langle \mathbf{v}, \mathbf{y}^{(i)} \rangle + k \mod p)$. The correctness of the private key $\mathtt{K}_{\mathbf{y}^{(i)}}$ is shown below.

$$\begin{aligned} & \mathtt{K}_1 \\ = \; & \langle \mathbf{s}, \mathbf{y}^{(i)} \rangle + k \mod p \\ = \; & \textstyle\sum_{j=1}^{\ell} s_j y_j^{(i)} + k \mod p \\ = \; & \textstyle\sum_{j=1}^{\ell} (u_j \cdot b + v_j) \cdot y_j^{(i)} + k \mod p \\ = \; & b \textstyle\sum_{j=1}^{\ell} u_j y_j^{(i)} + \sum_{j=1}^{\ell} v_j y_j^{(i)} + k \mod p \\ = \; & b \langle \mathbf{u}, \mathbf{y}^{(i)} \rangle + \langle \mathbf{v}, \mathbf{y}^{(i)} \rangle + k \mod p \\ = \; & \langle \mathbf{v}, \mathbf{y}^{(i)} \rangle + k \mod p. \end{aligned}$$

Challenge. Upon receiving $\mathbf{x}^*$, where $\langle \mathbf{x}^*, \mathbf{y}^{(i)} \rangle \neq 0$ for $i = 1, \ldots, q$, and two equal-lengthed messages M_0, M_1 from $\mathcal{A}$, the challenger $\mathcal{C}$ performs as follows.

1. Choose $\beta \in \{0,1\}$.
2. Choose $\delta \xleftarrow{\$} \mathbb{Z}_p$.
3. Set $\mathtt{C}'_0 = A'$, and $\widehat{\mathtt{C}}'_0 = A$.
4. For $i = 1$ to ℓ, compute $\mathtt{C}'_i = (C^{u_i} \cdot A^{v_i} \cdot g_T^{\delta x_i^*}) \cdot M_\beta$.
5. Set the challenge ciphertext $\mathtt{C}^* = (\mathtt{C}'_0, \widehat{\mathtt{C}}'_0, \mathtt{C}'_1, \mathtt{C}'_2, \ldots, \mathtt{C}'_\ell)$.
6. Return $\mathtt{C}^*$ to $\mathcal{A}$.

Here we implicitly set the randomness of the encryption procedure to a. Therefore, if $C = g_T^{ab}$, then we have $\mathtt{C}'_0 = aP, \widehat{\mathtt{C}}'_0 = g_T^a$, and for $i = 1, \ldots, \ell$,

$$\begin{aligned} \mathtt{C}'_i &= (C^{u_i} \cdot A^{v_i} \cdot g_T^{\delta x_i^*}) \cdot M_\beta \\ &= (g_T^{abu_i} \cdot g_T^{av_i} \cdot g_T^{\delta x_i^*}) \cdot M_\beta \\ &= (g_T^{a(bu_i + v_i)}) \cdot (g_T^{\delta x_i^*}) \cdot M_\beta \\ &= h_i^a \cdot g_T^{\delta x_i^*} \cdot M_\beta. \end{aligned}$$

That is, the challenge ciphertext $\mathtt{C}^*$ is a valid ciphertext.

Query Phase 2. This phase is the same as **Query Phase 1**.

Guess. The adversary $\mathcal{A}$ outputs a bit β'. The challenger $\mathcal{C}$ outputs 1 if $\mathcal{A}$ wins the game; outputs a random bit, otherwise.

Assume that the adversary $\mathcal{A}$ wins the game with advantage ε, i.e.,

$$\left| \Pr[\beta' = \beta] - \frac{1}{2} \right| \geq \varepsilon.$$

Note that, if $C = g_T^{ab}$, then the view of the adversary is identical as that in real world. Thus we have

$$\begin{aligned} & \Pr[\mathcal{C}(P, A', g_T, A, B, C = g_T^{ab}) = 1] \\ = \; & \Pr[\beta' = \beta] \\ \geq \; & \tfrac{1}{2} + \varepsilon. \end{aligned}$$

On the other hand, if C is a random element in $\mathbb{G}_T$, then the choice of β is independent from the adversary's view, and we have

$$\begin{aligned} & \Pr[\mathcal{C}(P, A', g_T, A, B, C \xleftarrow{\$} \mathbb{G}_T) = 1] \\ = \; & \Pr[\beta' = \beta] \\ = \; & \tfrac{1}{2}. \end{aligned}$$

Therefore, the advantage of $\mathcal{C}$ in solving the M-DDH$_{\mathbb{G}_T}$ problem is

$$\begin{aligned} & \left| \Pr[\mathcal{C}(P, A', g_T, A, B, C = g_T^{ab}) = 1] \right. \\ - \; & \left. \Pr[\mathcal{C}(P, A', g_T, A, B, C \xleftarrow{\$} \mathbb{G}_T) = 1] \right| \\ \geq \; & \left| (\tfrac{1}{2} + \varepsilon) - \tfrac{1}{2} \right| \\ \geq \; & \varepsilon. \end{aligned}$$

That means, if there is an adversary winning the game with non-advantage ε, there is an algorithm $\mathcal{C}$ solving the M-DDH$_{\mathbb{G}_T}$ problem with the probability greater than ε. □

5 COMPARISON

In this section, we compare the efficiency of the proposed IPE scheme with the previous works (Katz et al., 2008; Okamoto and Takashima, 2009; Attrapadung and Libert, 2010; Lewko et al., 2010; Okamoto and Takashima, 2011; Park, 2011; Okamoto and Takashima, 2012a,b; Kawai and Takashima, 2014; Zhenlin and Wei, 2015; Kim et al., 2016; Huang et al., 2016; Ramanna, 2016; Kurosawa and Phong, 2017; Xiao et al., 2017; Chen et al., 2018; Zhang et al., 2019)[‡], where the result is shown in Table 1. The comparison focuses on two parts, one is the private key length, and another is the number of pairing operations in the decryption algorithm. Since the efficiency of composite order bilinear groups is much lower than that of prime order groups, the order types of bilinear groups used in each scheme are also marked in the comparison table.

[‡]Since Attrapadung and Libert (2012), Okamoto and Takashima (2015) are the complete version of Attrapadung and Libert (2010), Okamoto and Takashima (2011), respectively, we only compare our work with Attrapadung and Libert (2010); Okamoto and Takashima (2011).

	Private Key Length	Number of Pairings for Decryption	Group Order
Katz et al. (2008)	$(2\ell+1)\|\mathbb{G}\|$	$2\ell+1$	Composite
Okamoto and Takashima (2009)	$(\ell+3)\|\mathbb{G}\|$	$\ell+3$	Prime
Attrapadung and Libert (2010)-1	$(\ell+1)\|\mathbb{G}\|$	2	Prime
Attrapadung and Libert (2010)-2	$(\ell+6)\|\mathbb{G}\|+(\ell-1)\|\mathbb{Z}_p\|$	9	Prime
Lewko et al. (2010)	$(2\ell+3)\|\mathbb{G}\|$	$2\ell+3$	Prime
Okamoto and Takashima (2011)-1	$(4\ell+1)\|\mathbb{G}\|$	9	Prime
Okamoto and Takashima (2011)-2	$9\|\mathbb{G}\|$	9	Prime
Okamoto and Takashima (2011)-3	$11\|\mathbb{G}\|$	11	Prime
Park (2011)	$(4\ell+2)\|\mathbb{G}\|$	$4\ell+2$	Prime
Okamoto and Takashima (2012a)	$(4\ell+2)\|\mathbb{G}\|$	$4\ell+2$	Prime
Okamoto and Takashima (2012b)-1	$(15\ell+5)\|\mathbb{G}\|$	$15\ell+5$	Prime
Okamoto and Takashima (2012b)-2	$(21\ell+9)\|\mathbb{G}\|$	$21\ell+9$	Prime
Kawai and Takashima (2014)	$6\ell\|\mathbb{G}\|$	6ℓ	Prime
Zhenlin and Wei (2015)	$\ell\|\mathbb{G}\|$	ℓ	Composite
Kim et al. (2016)	$3\|\mathbb{G}\|$	3	Prime
Huang et al. (2016)	$(4\ell+2)\|\mathbb{G}\|$	$4\ell+4$	Prime
Ramanna (2016)-1	$(2\ell+1)\|\mathbb{G}\|+(\ell-1)\|\mathbb{Z}_p\|$	3	Prime
Ramanna (2016)-2	$5\|\mathbb{G}\|$	3	Prime
Kurosawa and Phong (2017)	$2m\|\mathbb{G}\|$	$2m$	Prime
Xiao et al. (2017)	$(4\ell+5)\|\mathbb{G}\|$	$4\ell+5$	Prime
Chen et al. (2018)-1	$5\|\mathbb{G}\|$	5	Prime
Chen et al. (2018)-2	$7\|\mathbb{G}\|$	7	Prime
Zhang et al. (2019)	$(\ell+1)\|\mathbb{G}\|$	$\ell+1$	Composite
Ours	$1\|\mathbb{G}\|+1\|\mathbb{Z}_p\|$	1	Prime

Table 1: Efficiency Comparison. Here, ℓ denotes the vector length for an IPE scheme; $|\mathbb{Z}_p|$ and $|\mathbb{G}|$ denote the bit length of the representations for an element in $\mathbb{Z}_p$ and $\mathbb{G}$, respectively; m denotes the leakage-resilience parameter.

	Specification
OS	Ubuntu 18.04 LTS
CPU	Intel i7-4790 3.6GHz
RAM	8 gb
Language	Python 3.6
Library	Charm-Crypto v0.50

Table 2: The Environment of the Implementation.

One can observe that, in Table 1, our proposed scheme owns the shortest private key length and the smallest number of parings. Besides, both the private key length and the number of pairings in our proposed scheme are independent of the length of the predicate vector and the attribute vector. The most efficient existing scheme is Kim et al. (2016), where the private key length is three group elements and the three pairings are needed for decryption. In our scheme, the private key is only an element of $\mathbb{G}$ and an element of $\mathbb{Z}_p$, and only one pairing is necessary during decryption. One may also found that, in Kurosawa and Phong (2017), the private key length ($2m|\mathbb{G}|$) and the number of pairings ($2m$) are also independent of the length of the vectors, where m is the leakage-resilience parameter. However, m must at least greater or equal than 2. Therefore, the private key length and pairing number are still larger than those of ours.[§]

6 IMPLEMENTATION

We also implement our scheme and the schemes of Attrapadung and Libert (2012); Kim et al. (2016); Ramanna (2016), in order to show the efficiency comparison. The reason for choosing these three scheme is that,

- among all the existing IPE schemes, the first scheme of Attrapadung and Libert (2010) owns the smallest number of pairings for decryption (only 2 pairings required);
- among the schemes supporting constant private key length, the schemes of Kim et al. (2016); Ramanna (2016) own the smallest number of pairings for decryption (only 3 pairing required).

[§]The reason is that their scheme will degenerate to a conventional IPE scheme without leakage-resilience when $m = 1$.

	Encryption Time (ms)	Decryption Time (ms)	Private Key Length (kb)	Ciphertext Length (kb)
Attrapadung and Libert (2010)	100	100	31.7	0.937
Kim et al. (2016)	170	140	0.955	17.5
Ramanna (2016)	260	140	1.59	25.9
Ours	20	10	0.37	31.3

Table 3: The Implementation Result.

The environment of the implementation is shown in Table 2 and the implementation result is shown in Table 3. We implement these schemes by using the Charm-Crypto library Akinyele et al. (2013) via Python language. For schemes constructed over symmetric paring groups (Attrapadung and Libert (2010), ours), we choose the pairing group SS512 (Lee and Park, 2019) (a.k.a. type A groups); and the schemes constructed over asymmetric pairing groups ((Kim et al., 2016; Ramanna, 2016)), we choose the pairing group BN254 (Barreto and Naehrig, 2006) (a.k.a. type F groups). The SS512 groups is a super-singular elliptic curve group where the size of the base field order is 512 bits and the embedding degree is two. For a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ over the SS512 groups, the bit length of elements in $\mathbb{G}$ and $\mathbb{G}_T$ are 64 bytes and 128 bytes, respectively. In the case of the BN254 groups, the size of the base field order is 256 bits, and the embedding degree is 12. For a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ over the BN254 groups, the bit length of elements in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are 64 bytes, 128 bytes, and 384 bytes, respectively. We refer the readers to Lynn (2007) for more details. For the length of predicate and attribute vectors, we choose $\ell = 100$. From Table 3, one can observe that the encryption and decryption algorithm of our scheme are very efficient. For decryption (encryption), only 10 ms (20 ms) is required. Compared with Attrapadung and Libert (2010), Kim et al. (2016), Ramanna (2016), our encryption algorithm is 5x, 8.5x, and 13x faster than Attrapadung and Libert (2010), Kim et al. (2016), Ramanna (2016), respectively; and our decryption algorithm is 10x, 14x, 14x faster than Attrapadung and Libert (2010), Kim et al. (2016), Ramanna (2016), respectively. For the private key length, ours is also 86x, 2.6x, 4.3x shorter than Attrapadung and Libert (2010), Kim et al. (2016), Ramanna (2016), respectively. However, as a trade-off, the length of ciphertext in our scheme is the largest among these schemes.

7 CONCLUSION

This paper propose a practical inner product encryption scheme with constant-size private keys and constant pairing computations for decryption. More concretely, the private key of the proposed scheme has only an element in $\mathbb{G}$ and an element in $\mathbb{Z}_p$, and decryption requires only one pairing calculation. The security proof shows that our proposed scheme is co-selective IND-CPA secure under modified decisonal Diffie-Hellman assumption. Experimental results show that comparing with other schemes, our proposed scheme can effectively reduce the encryption and decryption time and private key length.

In future works, we will make our best effort to improve the efficiency of the ciphertext length, and provide the security proof for stronger security notions under standard assumptions.

Acknowledgment

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

REFERENCES

Agrawal, S., Boneh, D., and Boyen, X. (2010). Efficient Lattice (H)IBE in the Standard Model. In Gilbert, H., editor, *Advances in Cryptology – EUROCRYPT 2010*, pages 553–572, Berlin, Heidelberg. Springer Berlin Heidelberg.

Agrawal, S., Freeman, D. M., and Vaikuntanathan, V. (2011). Functional Encryption for Inner Product Predicates from Learning with Errors. In Lee, D. H. and Wang, X., editors, *Advances in Cryptology – ASIACRYPT 2011*, pages 21–40. Springer, Berlin, Heidelberg.

Akinyele, J. A., Garman, C., Miers, I., Pagano, M. W., Rushanan, M., Green, M., and Rubin, A. D. (2013). Charm: A Framework for Rapidly Prototyping Cryptosystems. *Journal of Cryptographic Engineering*, 3(2):111–128.

Apon, D., Fan, X., and Liu, F.-H. (2017). Vector Encoding over Lattices and Its Applications. *IACR Cryptology ePrint Archive*, 2017:455.

Attrapadung, N. (2014). Dual System Encryption via Doubly Selective Security: Framework, Fully Secure Functional Encryption for Regular Languages, and More. In Nguyen, P. Q. and Oswald, E., editors, *Advances in Cryptology – EUROCRYPT 2014*, pages 557–577. Springer, Berlin, Heidelberg.

Attrapadung, N. and Libert, B. (2010). Functional Encryption for Inner Product: Achieving Constant-Size Ciphertexts with Adaptive Security or Support for Negation. In Nguyen, P. Q. and Pointcheval, D., editors, *Public Key Cryptography – PKC 2010*, pages 384–402. Springer, Berlin, Heidelberg.

Attrapadung, N. and Libert, B. (2012). Functional Encryption for Public-attribute Inner Products: Achieving Constant-size Ciphertexts with Adaptive Security or Support for Negation. *J. Mathematical Cryptology*, 5(2):115–158.

Barreto, P. S. L. M. and Naehrig, M. (2006). Pairing-Friendly Elliptic Curves of Prime Order. In Preneel, B. and Tavares, S., editors, *Selected Areas in Cryptography*, pages 319–331, Berlin, Heidelberg. Springer Berlin Heidelberg.

Boneh, D. and Boyen, X. (2004). Efficient Selective-ID Secure Identity Based Encryption without Random Oracles. In *Proceedings of Eurocrypt 2004, volume 3027 of LNCS*, pages 223–238. Springer-Verlag.

Boneh, D., Boyen, X., and Goh, E.-J. (2005). Hierarchical Identity Based Encryption with Constant Size Ciphertext. In Cramer, R., editor, *Advances in Cryptology – EUROCRYPT 2005*, pages 440–456. Springer, Berlin, Heidelberg.

Boneh, D. and Franklin, M. (2001). Identity-Based Encryption from the Weil Pairing. In *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '01, pages 213–229, London, UK, UK. Springer-Verlag.

Boneh, D. and Waters, B. (2007). Conjunctive, Subset, and Range Queries on Encrypted Data. In Vadhan, S. P., editor, *Theory of Cryptography*, pages 535–554. Springer, Berlin, Heidelberg.

Chatterjee, S. and Mukherjee, S. (2019). Large Universe Subset Predicate Encryption based on Static Assumption (without Random Oracle). In *Cryptographers' Track at the RSA Conference*, pages 62–82. Springer.

Chen, J., Gong, J., and Wee, H. (2018). Improved Inner-Product Encryption with Adaptive Security and Full Attribute-Hiding. In Peyrin, T. and Galbraith, S., editors, *Advances in Cryptology - ASIACRYPT 2018*, pages 673–702, Springer, Cham.

Gentry, C., Sahai, A., and Waters, B. (2013). Homomorphic Encryption from Learning with Errors: Conceptually-simpler, Asymptotically-faster, Attribute-based. In *Annual Cryptology Conference*, pages 75–92. Springer.

Huang, S.-Y., Fan, C.-I., and Tseng, Y.-F. (2016). Enabled/Disabled Predicate Encryption in Clouds. *Future Generation Computer Systems*, 62:148 – 160.

K. Xagawa, K. (2013). Improved (Hierarchical) Inner-Product Encryption from Lattices. In Kurosawa, K. and Hanaoka, G., editors, *Public-Key Cryptography – PKC 2013*, pages 235–252. Springer, Berlin, Heidelberg.

Katz, J., Maffei, M., Malavolta, G., and Schröder, D. (2018). Subset Predicate Encryption and Its Applications. In Capkun, S. and Chow, S. S. M., editors, *Cryptology and Network Security*, pages 115–134, Cham. Springer International Publishing.

Katz, J., Sahai, A., and Waters, B. (2008). Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products. In Smart, N., editor, *Advances in Cryptology – EUROCRYPT 2008*, pages 146–162. Springer, Berlin, Heidelberg.

Kawai, Y. and Takashima, K. (2014). Predicate- and Attribute-Hiding Inner Product Encryption in a Public Key Setting. In Cao, Z. and Zhang, F., editors, *Pairing-Based Cryptography – Pairing 2013*, pages 113–130, Cham. Springer International Publishing.

Kim, I., Hwang, S. O., Park, J. H., and Park, C. (2016). An Efficient Predicate Encryption with Constant Pairing Computations and Minimum Costs. *IEEE Transactions on Computers*, 65(10):2947–2958.

Kurosawa, K. and Phong, L. T. (2017). Anonymous and Leakage Resilient IBE and IPE. *Designs, Codes and Cryptography*, 85(2):273–298.

Lee, K. (2017). Efficient Hidden Vector Encryptions and Its Applications. *CoRR*, abs/1702.07456.

Lee, K. and Park, J. H. (2019). Identity-Based Revocation From Subset Difference Methods Under Simple Assumptions. *IEEE Access*, 7:60333–60347.

Lewko, A., Okamoto, T., Sahai, A., Takashima, K., and Waters, B. (2010). Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In Gilbert, H., editor, *Advances in Cryptology – EUROCRYPT 2010*, pages 62–91. Springer, Berlin, Heidelberg.

Li, J., Zhang, D., Lu, X., and Wang, K. (2017). Compact (Targeted Homomorphic) Inner Product Encryption from LWE. In Qing, S., Mitchell, C., Chen, L., and Liu, D., editors, *International Conference on Information and Communications Security*, pages 132–140. Springer.

Lynn, B. (2007). *On the Implementation of Pairing-based Cryptosystems*. PhD thesis, Stanford University, Stanford University 450 Jane Stanford Way Stanford, CA 94305–2004.

Okamoto, T. and Takashima, K. (2009). Hierarchical Predicate Encryption for Inner-Products. In Matsui, M., editor, *Advances in Cryptology – ASIACRYPT 2009*, pages 214–231. Springer, Berlin, Heidelberg.

Okamoto, T. and Takashima, K. (2011). Achieving Short Ciphertexts or Short Secret-Keys for Adaptively Secure General Inner-Product Encryption. In Lin, D., Tsudik, G., and Wang, X., editors, *Cryptology and Network Security*, pages 138–159. Springer, Berlin, Heidelberg.

Okamoto, T. and Takashima, K. (2012a). Adaptively Attribute-Hiding (Hierarchical) Inner Product Encryption. In Pointcheval, D. and Johansson, T., ed-

itors, *Advances in Cryptology – EUROCRYPT 2012*, pages 591–608. Springer, Berlin, Heidelberg.

Okamoto, T. and Takashima, K. (2012b). Fully Secure Unbounded Inner-Product and Attribute-Based Encryption. In Wang, X. and Sako, K., editors, *Advances in Cryptology – ASIACRYPT 2012*, pages 349–366. Springer, Berlin, Heidelberg.

Okamoto, T. and Takashima, K. (2015). Achieving Short Ciphertexts or Short Secret-keys for Adaptively Secure General Inner-product Encryption. *Designs, Codes and Cryptography*, 77(2):725–771.

Park, J. H. (2011). Inner-Product Encryption under Standard Assumptions. *Designs, Codes and Cryptography*, 58(3):235–257.

Ramanna, S. C. (2016). More Efficient Constructions for Inner-Product Encryption. In Manulis, M., Sadeghi, A.-R., and Schneider, S., editors, *Applied Cryptography and Network Security*, pages 231–248. Springer, Cham.

Shamir, A. (1985). Identity-Based Cryptosystems and Signature Schemes. In *Proceedings of CRYPTO 84 on Advances in Cryptology*, pages 47–53, New York, NY, USA. Springer-Verlag New York, Inc.

Wang, Z., Fan, X., and Wang, M. (2018). Compact Inner Product Encryption from LWE. In Qing, S., Mitchell, C., Chen, L., and Liu, D., editors, *Information and Communications Security*, pages 141–153. Springer, Cham.

Waters, B. (2009). Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In *Annual International Cryptology Conference*, pages 619–636. Springer.

Xiao, S., Ge, A., Zhang, J., Ma, C., and Wang, X. (2017). Asymmetric Searchable Encryption from Inner Product Encryption. In Xhafa, F., Barolli, L., and Amato, F., editors, *Advances on P2P, Parallel, Grid, Cloud and Internet Computing*, pages 123–132. Springer, Cham.

Zhang, Y., Li, Y., and Wang, Y. (2019). Efficient Inner Product Encryption for Mobile Client with Constrained Capacity. *International Journal of Innovative Computing, Information and Control*, 15(1):209–226.

Zhenlin, T. and Wei, Z. (2015). A Predicate Encryption Scheme Supporting Multiparty Cloud Computation. In *2015 International Conference on Intelligent Networking and Collaborative Systems*, pages 252–256.