Practical Predicate Encryption for Inner Product

Yi-Fan Tseng, Zi-Yuan Liu* and Raylin Tso
Department of Computer Science, National Chengchi University, Taipei, Taiwan

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).

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 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?

*Corresponding author

Tseng, Y., Liu, Z. and Tso, R.
Practical Predicate Encryption for Inner Product.
DOI: 10.5220/0009804605530558
In *Proceedings of the 17th International Joint Conference on e-Business and Telecommunications (ICETE 2020) - SECRYPT*, pages 553-558
ISBN: 978-989-758-446-6

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.

2 PRELIMINARIES

2.1 Notations

Given a set S, "choose an element x randomly from the set S" will be denoted as "$x \overset{\$}{\leftarrow} S$". 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 $Z \overset{?}{=} g_T^{f(x_1,\ldots,x_n)}$. For an algorithm $\mathcal{A}$, the advantage of $\mathcal{A}$ in solving the (Q,R,f)-GDDHE problem is defined as*

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

Definition 2 (The Decisional Diffie-Hellman Problem over $\mathbb{G}_T$ (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 $a,b \overset{\$}{\leftarrow} \mathbb{Z}_p$, decide whether $C = g_T^{ab}$ or an random element from $\mathbb{G}_T$.*

Definition 3 (The Modified Decisional Diffie-Hellman Problem over $\mathbb{G}_T$ (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 $a,b \overset{\$}{\leftarrow} \mathbb{Z}_p$, decide whether $C = g_T^{ab}$ or an random element from $\mathbb{G}_T$.*

Theorem 1 (The Modified Decisional Diffie-Hellman Assumption over $\mathbb{G}_T$ (M-DDH$_{\mathbb{G}_T}$ assumption))**.** *We say that the M-DDH$_{\mathbb{G}_T}$ assumption holds if there is no algorithm $\mathcal{D}$ solving the M-DDH$_{\mathbb{G}_T}$ problem with a non-negligible advantage.*

Proof. Due to limited space, we give the proof in the full version of this paper Tseng et al. (2020). □

2.4 Definition of Inner Product Encryption

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

Setup$(1^\lambda, 1^\ell)$**.** Given the security parameters $(1^\lambda, 1^\ell)$, where $\lambda, \ell \in \mathbb{N}$, the algorithm outputs the system parameter $\mathtt{params}$ and the master secret key $\mathtt{msk}$. Note that the description of the attribute vector space $\mathfrak{A}$ and the predicate vector space $\mathfrak{P}$ will be implicitly included in $\mathtt{params}$. Besides, we require that the inner product operation over $\mathfrak{A}$ and $\mathfrak{P}$ should be well-defined.
Encrypt$(\mathtt{params}, \mathbf{x}, M)$**.** Given the system parameter $\mathtt{params}$, an attribute vector $\mathbf{x} \in \mathfrak{A}$, and a message M, the algorithm outputs a ciphertext $\mathtt{C}_\mathbf{x}$ for the attribute vector $\mathbf{x}$.
KeyGen$(\mathtt{params}, \mathtt{msk}, \mathbf{y})$**.** Given the system parameter $\mathtt{params}$, a predicate vector $\mathbf{y} \in \mathfrak{P}$, the algorithm outputs the private key $\mathtt{K}_\mathbf{y}$ for the predicate vector $\mathbf{y}$.
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 $\mathtt{K}_\mathbf{y}$, 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 $\mathtt{C}_\mathbf{x} \leftarrow$ **Encrypt**$(\mathtt{params}, \mathbf{x} \in \mathfrak{A}, M)$ and let $\mathtt{K}_\mathbf{y} \leftarrow$ **KeyGen**$(\mathtt{params}, \mathtt{msk}, \mathbf{y} \in \mathfrak{P})$, we have

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

where $(\mathsf{params}, \mathsf{msk}) \leftarrow$ **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 4 (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.*

We then present the co-selective security Attrapadung and Libert (2010); Attrapadung (2014) for inner product encryption. 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 $\mathbf{x}^* \in \mathfrak{A}$ where $\langle \mathbf{x}^*, \mathbf{y}^{(j)} \rangle \neq 0$ for $j = 1, \ldots, q$.
Definition 5 (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.*

3 THE PROPOSED INNER PRODUCT ENCRYPTION SCHEME

Our IPE scheme consists of four algorithms: **Setup**, **KeyGen**, **Encrypt**, and **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.

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

3.2 Security Proof

Theorem 2. *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. Due to limited space, we give the proof in the full version of this paper Tseng et al. (2020). □

4 COMPARISON AND IMPLEMENTATION

In this section, we compare the efficiency of the proposed IPE scheme with the previous works, 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.

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 environment of the implementation is shown in Table 2 and the implementation result is shown in Table 3. We note that due to limited space, please refer to the full version Tseng et al. (2020) for more comparison details and implementation details.

5 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

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.

	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 3: The Implementation Result.

	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 2: The Environment of the Implementation.

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

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.

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., 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 – ASI-*

ACRYPT 2011, pages 21–40. Springer, Berlin, Heidelberg.

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.

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 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.

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.

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., 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.

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.

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., editors, *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.

Tseng, Y.-F., Liu, Z.-Y., and Tso, R. (2020). Practical Predicate Encryption for Inner Product. Cryptology ePrint Archive, Report 2020/270.

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.

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.