International Journal of Innovative Computing, Information and Control
Volume **15**, Number **1**, February **2019**

ICIC International ©2019 ISSN 1349-4198
pp. **209–226**

EFFICIENT INNER PRODUCT ENCRYPTION FOR MOBILE CLIENTS WITH CONSTRAINED COMPUTATION CAPACITY

Yu Zhang[1], Yin Li[1] and Yifan Wang[2]

[1]School of Computer and Information Technology
Xinyang Normal University
No. 237, Nanhu Road, Xinyang 464000, P. R. China
willow1223@126.com; yufeiyangli@gmail.com

[2]Wayne State University
42 W Warren Ave, Detroit, MI 48202, United States
yifan.wang@wayne.edu

Received March 2018; revised July 2018

Abstract. *Predicate encryption is an encryption paradigm in which a secret key associated with a predicate f can decrypt a ciphertext corresponding to an attribute I when $f(I) = 1$. In the predicate encryption domain, inner-product based encryption plays an important role and has been studied for a long time. Earlier schemes were proven to be selectively secure. Recently, to obtain a higher security level, it has been adjusted to a fully attribute-hiding model as well as proven to be adaptively secure. However, the usage of the dual pairing vector space led to the quadratic growth of the time cost of the encryption and key generation. This characteristic makes it difficult to apply in some resource constrained environment, e.g., mobile networks. In this paper, by using dual system encryption technology combined with mathematical structure of the composite bilinear order group, we propose an adaptive secure inner product encryption scheme with less encryption and key generation time cost. The theoretical analysis and experiment results show that our scheme has lower space and time complexities. This merit makes our proposal more suitable for the mobile cloud environment compared with the state-of-the-art schemes.*

Keywords: Public key system, Pairing-based cryptography, Inner product encryption, Predicate encryption, Function encryption

1. **Introduction.**

In a traditional public key encryption system, a sender encrypts a message m and obtains a ciphertext by using the public key PK. Only the owner of the secret key SK can decrypt the ciphertext, and cannot learn any information about the ciphertext before decrypting it. However, it is not always the case. For example, a person holding SK only wants to decrypt the specific ciphertext which contains special attributes. The traditional public key encryption system fails to support it since the system cannot detect whether the special attributes are contained in the ciphertext or not without decrypting the ciphertext. Predicate encryption (PE) is a new way to address the problem of how to estimate whether a ciphertext is associated with some attributes without decrypting this ciphertext. In a PE scheme, a ciphertext associated with an attribute I in a set Σ can be decrypted by a secret key corresponding to the predicate f if and only if $f(I) = 1$.

Inner product encryption (IPE) is a special case of PE. In an IPE scheme, each ciphertext associated with an attribute vector $\vec{x}$ can be decrypted by a secret key corresponding to predicate vector $\vec{v}$ if and only if $f_{\vec{v}}(\vec{x}) = 1$, which means that $\vec{v} \cdot \vec{x} = 0$. This scheme

DOI: 10.24507/ijicic.15.01.209

can be applied in many applications, such as searchable encryption [1, 2, 3, 4]. The first IPE scheme was proposed in [5]. It enables more complex evaluations on disjunctions, polynomials, and conjunctive/disjunctive normal form (CNF/DNF) formulae. However, this scheme was proven to be selective security. By using the technology called dual system encryption presented in [6, 7], an adaptively secure and weakly attribute-hiding IPE scheme was presented in [8]. Later, an IPE scheme which was proven to be adaptively secure in a fully attribute-hiding model was proposed in [9]. Due to the usage of dual pairing vector space (DPVS) [8, 9], the time consuming of both encryption and key generation in these two adaptively secure IPE schemes are linear with $O(n^2)$, where n is the size of vector. Therefore, it is necessary to construct a more efficient IPE scheme in time complexity.

In this paper, we focus on constructing a new IPE scheme that can improve the encryption and key generation efficiency. The contributions are summarized as follows.

1) Using the composite order bilinear group and dual encryption system, we propose a new IPE scheme which is proven to be adaptively secure in a weakly attribute-hiding model. Although composite order bilinear group is less efficient than the prime order one, doing so we can totally discard the awfully time-consuming DPVS technology with only a very little sacrifice of security. Therefore, our scheme needs less time to encrypt message with a predicate vector and create public key (PK), master secret key (MSK) and secret key with an attribute vector.
2) The experiment result shows that the encryption and key generation operations preformed on the client device in our scheme are more efficient than that in other schemes [5, 8, 9]. Besides, the time and space cost of creating PK and MSK in our scheme is linear with n, while that in [8, 9] is linear with n^2, where n is the number of dimensions in a predicate/attribute vector. Since client device is always a mobile device with less storage space and limited computation capacity, compared with the previous IPE scheme, the proposed scheme is much more suitable for the resource constrained environment.

Organization. This paper is organized as follows. In Section 2, we first introduce some works related to our proposal; then the IPE model and its security model are given; finally, a brief introduction of bilinear groups is presented and the complexity assumptions used to prove the security of our scheme are also stated. In Section 3, we propose the new IPE scheme. The security proofs and the performance analysis of our scheme are presented in Section 4 and Section 5, respectively. The conclusions are presented in Section 6.

2. **Preliminaries.**

In this section, we first introduce some works related to IPE. After that, we define the framework of an IPE scheme, and give the formal security model we will use. Moreover, we give required concepts on composite order bilinear group and complexity assumptions. Additionally, in order to express these concepts clearly, the important notations used in this paper are introduced in Table 1.

2.1. **Related work.**

The IPE scheme is a particular case of the PE prototype, which stems from identity-based encryption (IBE), hierarchical-IBE (HIBE), attribute-based encryption (ABE) and searchable encryption (SE).

IBE. In an IBE scheme, an authority generates and distributes keys to users with associated identities, and each user takes advantage of their own identity to encrypt the message. Note that the encrypted message can only be decrypted with the correct identity. The earlier IBE schemes were constructed by Boneh and Franklin [10] and Cocks [11]. The improved IBE schemes were proposed in [12, 13].

Table 1. Notations

Notation	Description
$\vec{x}$	Attribute vector
$\vec{v}$	Predicate vector
PK	Public key
MSK	Master secret key
m	Message
$sk_{\vec{v}}$	Secret key of $\vec{v}$
C	Ciphertext
n	The length of $\vec{x}$
q	The number of the secret keys which adversary queries
i	A counter, where $i \in [1, n]$
j	A counter, where $j \in [1, q]$

HIBE [14, 15] aims for applying a hierarchical structure on identities to the IBE system, which enable a user holding a superior identity to generate and delegate secret keys associated with its subordinate identities. The following works focus on the issue of enhancing the security level and efficiency [6, 7, 16].

ABE. ABE was first introduced by Sahai and Waters [17], and can be classified into ciphertext-policy ABE (CP-ABE) and key-policy ABE (KP-ABE) [18]. In a CP-ABE scheme, each key is associated with a group of attributes and each ciphertext corresponds with an access control policy, while the situation is reversed in a KP-ABE scheme. The decryption can only be accomplished in the situation where the attribution meets the access control policy. After that, many researchers try their best to construct secure and efficient CP-ABE and KP-ABE schemes introduced in [19, 20, 21].

SE. SE can be regarded as the anonymous-IBE (AIBE) introduced in [22, 23], and was proposed to realize keyword searching over encrypted data with various search functions. There are two main categories of SE according to its applications: searchable public key encryption and searchable symmetric key encryption. Over the last few years, many searchable SE schemes have been proposed, which achieve complex search conditions such as Boolean keyword search, personal keyword search, and query result ranking [1, 2, 3, 4].

PE and IPE. There are two common cases in the PE domain. One is called hidden vector encryption (HVE) firstly introduced by [24] and extended in [25, 26], and the other is IPE firstly proposed by Katz et al. [5]. The first IPE scheme was proven to be the selective security. In order to construct an adaptively secure scheme, by using a DPVS methodology, an adaptively secure and weakly attribute-hiding IPE scheme was presented in [8]. After that, an IPE scheme which was proven to be adaptively secure in a fully attribute-hiding model was proposed in [9]. In this work, a variant IPE scheme with a constant size of SK is also proposed, in which the predicate vector must be disclosed as an additional information. The work given in [27] improved the efficiency of the variant IPE scheme. In recent two years, under the private-key setting, two IPE works supporting multi-inputs have been proposed to accelerate decryption process [28, 29]. In addition, a private-key IPE scheme with strongly full-hiding property was proposed in [30].

2.2. **Framework of IPE.** The original definition of IPE was presented in [5]. Specifically, for the class of inner-product predicate, an attribute can be expressed as a vector $\vec{x}$ and a predicate, associated with a vector $\vec{v}$, can be expressed as $f_{\vec{v}}$. We have $f_{\vec{v}}(\vec{x}) = 1$, if and only if $\vec{v} \cdot \vec{x} = 0$. We denote Σ as an arbitrary set of attributes and $\mathbb{F}$ as an arbitrary set of predicates over Σ.

Definition 2.1. [5] *An IPE scheme with predicates $\mathbb{F}$ and attributes Σ consists of four probabilistic polynomial-time algorithms: **Setup**, **KeyGen**, **Enc** and **Dec**. They are given as follows.*

*1) **Setup** takes as input the security parameter 1^n, and it outputs PK and MSK.*

*2) **KeyGen** takes as input the master secret key MSK and the predicate vector $\vec{v} \in \mathbb{F}$. It outputs the corresponding secret key $sk_{\vec{v}}$.*

*3) **Enc** takes as input the public key PK, the message m, and the attribute vector $\vec{x} \in \Sigma$. It returns the ciphertext C.*

*4) **Dec** takes as input the public key PK, the secret key $sk_{\vec{v}}$ and the ciphertext C. It outputs either the message m or a symbol $\perp$.*

***Consistency in IPE**: For all $f_{\vec{v}} \in \mathbb{F}$ and $\vec{x} \in \Sigma$, for correctly generated **Setup**$(1^n) \to \{PK, MSK\}$, **KeyGen**$(MSK, \vec{v}) \to sk_{\vec{v}}$ and $Enc(PK, m, \vec{x}) \to C$, it holds that $m =$ **Dec**$(PK, sk_{\vec{v}}, C)$ if $\vec{v} \cdot \vec{x} = 0$. Otherwise, it outputs $\perp$.*

2.3. **Security model for IPE.** An IPE scheme must ensure attribute-hiding security. The security definition of IPE called "adaptively secure and weakly attribute-hiding" is described as follows.

Definition 2.2. [8] *An IPE scheme is adaptively secure and weakly attribute-hiding against the chosen plaintext attacks, if for all probabilistic polynomial-time adversaries $\mathbb{A}$, the advantage of $\mathbb{A}$ in the following experiment is negligible.*

*1) **Setup**(1^n) is run to generate PK and MSK, and PK is given to the adversary $\mathbb{A}$.*

2) $\mathbb{A}$ may adaptively make q' secret key queries for q' predicate vectors $\vec{v_1}, \vec{v_2}, \ldots, \vec{v_{q'}}$. In response, $\mathbb{A}$ is given the corresponding keys $sk_{\vec{v_1}}, sk_{\vec{v_2}}, \ldots, sk_{\vec{v_{q'}}}$.

3) $\mathbb{A}$ randomly outputs two challenge attribute vectors $\vec{x}^{(0)}$, $\vec{x}^{(1)}$ and two challenge messages $m^{(0)}$, $m^{(1)}$, subject to the following restrictions: for the secret key of the predicate vector $\vec{v_l}$, there are $\vec{v_l} \cdot \vec{x}^{(0)} \neq 0$ and $\vec{v_l} \cdot \vec{x}^{(1)} \neq 0$, where $l \in [1, q']$.

*4) A random bit b is chosen, and $\mathbb{A}$ is given $C^{(b)} \to$**Enc**$\left(PK, m^{(b)}, \vec{x}^{(b)}\right)$.*

5) The adversary $\mathbb{A}$ may continue to request keys corresponding to the additional predicates vectors, $v_{q'\vec{+}1}, v_{q'\vec{+}2}, \ldots, \vec{v_q}$, subject to the restriction given in Step 3). $\mathbb{A}$ is given the corresponding keys $sk_{v_{q'\vec{+}1}}, sk_{v_{q'\vec{+}2}}, \ldots, sk_{\vec{v_q}}$.

6) $\mathbb{A}$ outputs a bit $b^{'}$, and succeeds if $b' = b$.

The advantage of the adversary $\mathbb{A}$ in breaking the PE supporting inner product scheme is defined as $ADV_{\mathbb{A}}^{PE} = \left|Pr[b' = b] - \frac{1}{2}\right|$.

2.4. **Composite order bilinear groups.** Composite order bilinear groups were first used in cryptographic construction in [31]. We use a group with order N which is the product of four distinct primes and a generator $\mathfrak{g}$ which takes as input a security parameter 1^n and outputs a description $I = (p_1, p_2, p_3, p_4, G, G_T, \hat{e})$, where p_1, p_2, p_3, p_4 are primes, G and G_T are cyclic groups of order $N = p_1p_2p_3p_4$, and $\hat{e} : G \times G \to G_T$ is a non-degenerate bilinear map with the properties as follows:

1) Bilinear: $\hat{e}\left(u^a, v^b\right) = \hat{e}(u, v)^{ab}$, where $u, v \in G$ and $a, b \in Z_N^*$;

2) Non-degenerate: $\hat{e}$ does not send all pairs of points in $G \times G$ to the identity in G_T. If g is a generator of G then $\hat{e}(g, g)$ is a generator of G_T;

3) Computable: there is an efficient algorithm to compute $\hat{e}(u, v)$, for any $u, v \in G$.

We further require that the group operations in G and G_T, as well as the bilinear map $\hat{e}$, are computable in deterministic polynomial time with respect to n. Let G_{p1}, G_{p2}, G_{p3} and G_{p4} denote the subgroups of G having order p_1, p_2, p_3 and p_4 respectively. For $a, b, c \in \{1, p_1, p_2, p_3, p_4\}$, suppose that $G_1 = 1$, it can be found that $G_{abc} = G_a \times G_b \times G_c$

where the subgroup of G with order x is denoted by G_x. Furthermore, suppose that $h_1 \in G_\alpha$ and $h_2 \in G_\beta$, we can verify that $\hat{e}(h_1, h_2) = 1$ if $gcd(\alpha\beta|N^2, N) = N$. This is called the orthogonality property and is a crucial tool in our construction.

For proving the security of our construction, we give the complexity assumptions presented in [12, 23] as follows.

2.5. Complexity assumptions.

Assumption 2.1. *Given a group generator* $\mathfrak{g}$*, we define the following distribution:*

$\mathbb{G} = (N = p_1p_2p_3p_4, G, G_T, \hat{e}) \overset{R}{\leftarrow} \mathfrak{g}$ $g_1 \overset{R}{\leftarrow} G_{p_1}$, $g_3 \overset{R}{\leftarrow} G_{p_3}$, $g_4 \overset{R}{\leftarrow} G_{p_4}$

$D = (G, g_1, g_3, g_4)$ $T_1 \overset{R}{\leftarrow} G_{p_1p_2}$, $T_2 \overset{R}{\leftarrow} G_{p_1}$

The advantage of an algorithm $\mathbb{A}$ in breaking Assumption 2.1 is defined by:

$$Adv1_{\mathfrak{g},\mathbb{A}}(n) = Pr[\mathbb{A}(D, T_1) = 1] - Pr[\mathbb{A}(D, T_2) = 1] \tag{1}$$

Definition 2.3. *For any probabilistic polynomial-time algorithm* $\mathbb{A}$*, if* $Adv1_{\mathfrak{g},\mathbb{A}}(n)$ *is a negligible function of* n*, we can say that Assumption 2.1 holds for the generator* $\mathfrak{g}$.

In addition, suppose that $T_1 \overset{R}{\leftarrow} G_{p_3p_2}$, $T_2 \overset{R}{\leftarrow} G_{p_3}$ or $T_1 \overset{R}{\leftarrow} G_{p_4p_2}$, $T_2 \overset{R}{\leftarrow} G_{p_4}$, we can find that Assumption 2.1 is still correct.

Assumption 2.2. *Given a group generator* $\mathfrak{g}$*, we define the following distribution:*

$\mathbb{G} = (N = p_1p_2p_3p_4, G, G_T, \hat{e}) \overset{R}{\leftarrow} \mathfrak{g}$

$g_1, D_1 \overset{R}{\leftarrow} G_{p_1}$, $D_2, B_2 \overset{R}{\leftarrow} G_{p_2}$, $B_3, g_3 \overset{R}{\leftarrow} G_{p_3}$, $g_4 \overset{R}{\leftarrow} G_{p_4}$

$D = (G, g_1, g_3, g_4, D_1D_2, B_2B_3)$ $T_1 \overset{R}{\leftarrow} G_{p_1p_2p_3}$, $T_2 \overset{R}{\leftarrow} G_{p_1p_3}$

The advantage of an algorithm $\mathbb{A}$ in breaking Assumption 2.2 is defined by:

$$Adv2_{\mathfrak{g},\mathbb{A}}(n) = Pr[\mathbb{A}(D, T_1) = 1] - Pr[\mathbb{A}(D, T_2) = 1] \tag{2}$$

Definition 2.4. *For any probabilistic polynomial-time algorithm* $\mathbb{A}$*, if* $Adv1_{\mathfrak{g},\mathbb{A}}(n)$ *is a negligible function of* n*, we can say that Assumption 2.2 holds for the generator* $\mathfrak{g}$.

Assumption 2.3. *Given a group generator* $\mathfrak{g}$*, we define the following distribution:*

$\mathbb{G} = (N = p_1p_2p_3p_4, G, G_T, \hat{e}) \overset{R}{\leftarrow} \mathfrak{g}$

$\alpha, s, r \overset{R}{\leftarrow} Z_N$, $g_1 \overset{R}{\leftarrow} G_{p_1}$, $g_2, A_2, B_2 \overset{R}{\leftarrow} G_{p_2}$, $g_3 \overset{R}{\leftarrow} G_{p_3}$, $g_4 \overset{R}{\leftarrow} G_{p_4}$

$D = (G, g_1, g_2, g_3, g_4, g_1^\alpha A_2, g_1^s B_2, g_2^r, A_2^r)$

$T_1 = \hat{e}(g_1, g_1)^{\alpha s}$, $T_2 \overset{R}{\leftarrow} G_T$

The advantage of an algorithm $\mathbb{A}$ in breaking Assumption 2.3 is defined by:

$$Adv3_{\mathfrak{g},\mathbb{A}}(n) = Pr[\mathbb{A}(D, T_1) = 1] - Pr[\mathbb{A}(D, T_2) = 1] \tag{3}$$

Definition 2.5. *For any probabilistic polynomial-time algorithm* $\mathbb{A}$*, if* $Adv1_{\mathfrak{g},\mathbb{A}}(n)$ *is a negligible function of* n*, we can say that Assumption 2.3 holds for the generator* $\mathfrak{g}$.

Assumption 2.4. *Given a group generator* $\mathfrak{g}$*, we define the following distribution:*

$\mathbb{G} = (N = p_1p_2p_3p_4, G, G_T, \hat{e}) \overset{R}{\leftarrow} \mathfrak{g}$

$E_1E_2 \overset{R}{\leftarrow} G_{p_1p_2}$, $A_1A_4 \overset{R}{\leftarrow} G_{p_1p_4}$, $g_2 \overset{R}{\leftarrow} G_{p_2}$, $g_3 \overset{R}{\leftarrow} G_{p_3}$, $g_4 \overset{R}{\leftarrow} G_{p_4}$

$D = (G, g_2, g_3, g_4, A_1A_4, E_1E_2)$

$T_1 \overset{R}{\leftarrow} G_{p_1p_2p_4}$, $T_2 \overset{R}{\leftarrow} G_{p_2p_4}$

The advantage of an algorithm $\mathbb{A}$ in breaking Assumption 2.4 is defined by:

$$Adv4_{\mathfrak{g},A}(n) = Pr[A(D, T_1) = 1] - Pr[A(D, T_2) = 1] \tag{4}$$

Definition 2.6. *For any probabilistic polynomial-time algorithm $\mathbb{A}$, if $Adv1_{\mathfrak{g},\mathbb{A}}(n)$ is a negligible function of n, we can say that Assumption 2.4 holds for the generator $\mathfrak{g}$.*

Assumption 2.4 is based on the general subgroup decision (GSD) assumption presented in [12].

3. **Construction for IPE.**

We now present our core IPE scheme, which is based on the composite order bilinear group, and can be proven fully secure using the dual system encryption methodology. This scheme we present here is related to the schemes based on the dual pairing vector space (DPVS) framework in prime order bilinear groups. The intuition behind the core of the construction is applying a composite order bilinear group instead of using the DPVS method that needs more encryption time, where the randomness of the ciphertext and the key depends on two different sub-order groups.

Specifically, the IPE scheme we constructed works as follows.

Setup($\mathbf{1^n}$) The algorithm chooses a bilinear group G of order $N = p_1p_2p_3p_4$ (where p_1, p_2, p_3, p_4 are distinct primes). Let G_{p_i} denote the subgroup of order p_i in G. Randomly choosing $\alpha \in Z_N$, $\alpha_i \in Z_N$, $\beta_i \in Z_N$, $U_1 \in G_{p_1}$, $A_1 \in G_{p_1}$, $B_1 \in G_{p_1}$, $A_{4i} \in G_{p_4}$, $B_{4i} \in G_{p_4}$, $U_4 \in G_{p_4}$ and $g_4 \in G_{p_4}$, where $i \in [1, n]$, PK is published as:

$$PK = \left\{N, U_1U_4, {A_1}^{\beta_i}A_{4i}, {A_1}^{\alpha_i}B_{4i}, g_4, \hat{e}(B_1, U_1)^{\alpha}\right\} \tag{5}$$

The master secret key is

$$MSK = \{\alpha, \alpha_i, \beta_i, U_1, A_1, B_1, g_3\} \tag{6}$$

where g_3 is a generator of G_{p_3}.

Enc($\boldsymbol{m}, \boldsymbol{\vec{x}}, \boldsymbol{PK}$) Choosing $n+2$ random elements $s, c_1, c_2, \ldots, c_n, c_{n+1} \in Z_N$. For the vector $\vec{x} = \{x_1, x_2, \ldots, x_n\}$, the encryption algorithm creates the ciphertext

$$C = (C_0, C_{11}, C_{12}, \ldots, C_{1n}, C_2) \tag{7}$$

where

$$C_0 = m\hat{e}(B_1, U_1)^{s\alpha} \tag{8}$$

$$C_{1i} = \left({A_1}^{\beta_i}A_{4i}\right)^{sx_i} \times \left({A_1}^{\alpha_i}B_{4i}\right)^s \times {g_4}^{c_i} = {A_1}^{s(\beta_i x_i + \alpha_i)}C_{4i} \tag{9}$$

$$C_2 = (U_1U_4)^s \times g_4^{c_{n+1}} = U_1^sC_4 \tag{10}$$

In Equation (9) and Equation (10), $C_{4i} = A_{4i}^{sx_i}B_{4i}^s{g_4}^{c_i}$ and $C_4 = g_4^{c_{n+1}}U_4^s$ respectively.

KeyGen($\boldsymbol{\vec{v}}, \boldsymbol{MSK}$) The key generation algorithm chooses $r \in Z_N$ and generates random elements R_{3i}, R_3 by using g_{p_3} and raising it to the random exponents modulo N. For a vector $\vec{v} = \{v_1, v_2, \ldots, v_n\}$,

$$sk_{\vec{v}} = (K_{11}, K_{12}, \ldots, K_{1n}, K_2) \tag{11}$$

where $K_{1i} = {U_1}^{r\frac{v_i}{\beta_i}}R_{3i}$, $K_2 = {B_1}^{\alpha}{A_1}^{r\sum_{i=1}^{n}\frac{\alpha_i v_i}{\beta_i}}R_3$.

Dec($\boldsymbol{C}, \boldsymbol{sk_{\vec{v}}}, \boldsymbol{PK}$) The algorithm computes

$$K_0 = \frac{\hat{e}(C_2, K_2)}{\prod_{i=1}^{n}\hat{e}(C_{1i}, K_{1i})}. \tag{12}$$

If $\vec{x} \cdot \vec{v} = 0$, the plaintext m can be required by computing $\frac{C_0}{K_0}$.

Correctness. According to Equation (7) to Equation (11), we have

$$\begin{aligned} K_0 &= \frac{\hat{e}(C_2, K_2)}{\prod_{i=1}^n \hat{e}(C_{1i}, K_{1i})} \\ &= \frac{\hat{e}\left({U_1}^s C_4, {B_1}^\alpha {A_1}^{r\sum_{i=1}^n \frac{\alpha_i v_i}{\beta_i}} R_3\right)}{\prod_{i=1}^n \hat{e}\left({A_1}^{s(\beta_i x_i + \alpha_i)} C_{4i}, U_1^{r\frac{v_i}{\beta_i}} R_{3i}\right)} \\ &= \frac{\hat{e}(B_1, U_1)^{s\alpha} \hat{e}(A_1, U_1)^{rs\sum_{i=1}^n \frac{\alpha_i v_i}{\beta_i}}}{\prod_{i=1}^n \hat{e}(A_1, U_1)^{rs\frac{\alpha_i v_i}{\beta_i}} \hat{e}(A_1, U_1)^{rsx_i v_i}} \\ &= \frac{\hat{e}(B_1, U_1)^{s\alpha}}{\hat{e}(A_1, U_1)^{rs\sum_{i=1}^n x_i v_i}} \end{aligned} \tag{13}$$

Obviously, if $\vec{x} \cdot \vec{v} = 0$, then $\frac{C_0}{K_0} = m$.

4. **Security Analysis.** To prove the security of our IPE system, we present the construction of the semi-functional key and the semi-functional ciphertext in our scheme. These constructions will just be used in the proof rather than in the real PE system.

Semi-functional Ciphertext Let g_2 be a generator of the subgroup G_{p_2}. A semi-functional ciphertext is created as follows: A normal ciphertext $\{C'_0, C'_{11}, C'_{12}, \dots, C'_{1n}, C'_2\}$ is constructed by the encryption algorithm. Choosing $n+2$ random elements $x, z_{c1}, z_{c2}, \dots, z_{cn}$ and $z_c \in Z_N$, C_0 is set to be C'_0, C_{1i} is set to be $C'_{1i} g_2^{xz_{ci}}$ for each $i \in [1, n]$ and C_2 is set to be $C'_2 g_2^{xz_c}$. The semi-functional ciphertext is

$$\{C_0, C_{11}, C_{12}, \dots, C_{1n}, C_2\} \tag{14}$$

Semi-functional Key Let g_2 be a generator of the subgroup G_{p_2}. A semi-functional key is created as follows: A normal key $\{K'_{11}, K'_{12}, \dots, K'_{1n}, K'_2\}$ is constructed by the encryption algorithm. Choosing $n+2$ random elements $\xi, z_{k1}, z_{k2}, \dots, z_{kn}$ and $z_k \in Z_N$, K_{1i} is set to be $K'_{1i} g_2^{\xi z_{ki}}$ for each $i \in [1, n]$ and K_2 is set to be $K'_2 g_2^{\xi z_k}$. The semi-functional key is

$$\{K_{11}, K_{12}, \dots, K_{1n}, K_2\} \tag{15}$$

When using the semi-functional key to decrypt the semi-functional ciphertext, an additional factor

$$\hat{e}(g_2, g_2)^{x\xi\left(z_c z_k - \sum_{i=1}^n z_{ci} z_{ki}\right)} \tag{16}$$

is generated. Obviously, if $z_c z_k = \sum_{i=1}^n z_{ci} z_{ki} \bmod p_2$, the decryption will still work.

The security proof of our PE scheme relies on the four assumptions presented in Section 2. The security is proven by using a hybrid method which uses a sequence of games. These games are described as follows.

1) $Game_{Real}$ This game is the real security game.
2) $Game_{Restricted}$ This game is the same as the real game except that the attacker cannot ask for keys for the vector $\vec{v} = (v_1, v_2, \dots, v_n)$ satisfying $\vec{x} \cdot \vec{v} = \sum_{i=1}^n x_i v_i = 0 \bmod p_2$, where $\vec{x} = (x_1, x_2, \dots, x_n)$ is one of the challenge vectors. We will retain this stronger restriction throughout the subsequent games.
3) $Game_k$ For each $k \in [0, q]$, we define $Game_k$ which is similar to $Game_{Restricted}$ except that the ciphertext given to $\mathbb{A}$ is semi-functional and the first k keys are semi-functional. The rest of the keys are normal. In $Game_0$, all keys given to $\mathbb{A}$ are normal and the ciphertext is semi-functional. In $Game_q$, the ciphertext and all of the keys are semi-functional.

4) $Game_{Final_0}$ This game is the same as $Game_q$ except that the ciphertext is a semi-functional encryption of a random message. This message is not one of the two messages which are requested by the attacker $\mathbb{A}$.
5) $Game_{Final_{1,k}}$ This game is the same as $Game_{Final_0}$ except that the ciphertext is a semi-functional encryption of a challenge vector where its first k elements are random and the rest elements are normal.

It can be seen that $Game_{Final_{1,0}}$ is the same as $Game_{Final_0}$ and $Game_{Final_{1,n}}$ is a game such that the ciphertext is a semi-functional encryption of a random message and a random vector. The key point for proving the security of our scheme is to show these games are indistinguishable in the following lemmas.

Lemma 4.1. *Suppose that there exists a probabilistic polynomial time (PPT) algorithm $\mathbb{A}$ such that $Adv^{\mathbb{A}}_{Game_{Real}} - Adv^{\mathbb{A}}_{Game_{Restricted}} = \epsilon$, a PPT algorithm $\mathbb{B}$ with advantage $\geq \frac{\epsilon}{3}$ in breaking either Assumption 2.1 or Assumption 2.2 can be built.*

Lemma 4.2. *Suppose that there exists a PPT algorithm $\mathbb{A}$ such that $Adv^{\mathbb{A}}_{Game_{Restricted}} - Adv^{\mathbb{A}}_{Game_0} = \epsilon$, a PPT algorithm $\mathbb{B}$ with advantage ϵ in breaking Assumption 2.1 can be built.*

Lemma 4.3. *Suppose that there exists a PPT algorithm $\mathbb{A}$ such that $Adv^{\mathbb{A}}_{Game_{k-1}} - Adv^{\mathbb{A}}_{Game_k} = \epsilon$, a PPT algorithm $\mathbb{B}$ with advantage ϵ in breaking Assumption 2.2 can be built.*

Lemma 4.4. *Suppose that there exists a PPT algorithm $\mathbb{A}$ such that $Adv^{\mathbb{A}}_{Game_q} - Adv^{\mathbb{A}}_{Game_{Final_0}} = \epsilon$, a PPT algorithm $\mathbb{B}$ with advantage ϵ in breaking Assumption 2.3 can be built.*

Lemma 4.5. *Suppose that there exists a PPT algorithm $\mathbb{A}$ such that $Adv^{\mathbb{A}}_{Game_{Final_{1,k-1}}} - Adv^{\mathbb{A}}_{Game_{Final_{1,k}}} = \epsilon$, a PPT algorithm $\mathbb{B}$ with advantage ϵ in breaking Assumption 2.4 can be built.*

The proofs of Lemmas 4.1-4.5 are given in Appendix.

Theorem 4.1. *If Assumptions 2.1, 2.2, 2.3, and 2.4 hold, then our PE system is secure.*

Proof: If Assumptions 2.1, 2.2, 2.3, and 2.4 hold, the real security game is indistinguishable from $Game_{Final_{1,n}}$ based on the previous lemmas. In $Game_{Final_{1,n}}$, the value of β is information-theoretically hidden from the attacker. Hence, we can say that the attacker can attain no advantage in breaking our PE system. □

5. **Performance Evaluation.**

5.1. **Theoretical analysis.**

We denote IPE schemes proposed in [5, 8, 9] by KSW08, LOSTW10 and OT12 respectively, and compare the proposed scheme with these schemes. The reason why we choose these schemes is that KSW08, OT12 are the best schemes in terms of time space complexity and security respectively while LOSTW10 is a trade-off scheme between security and performance.

Let $|T_e|$, $|T_{3e}|$ and $|T_{4e}|$ be the time cost for a pairing operation [31] on G, $G3$ and $G4$, and $|T_G|$, $|T_{3G}|$ and $|T_{4G}|$ be the time cost for the exponentiation operation on G, $G3$ and $G4$, where G, $G3$, and $G4$ are a group of a prime order, a composite group of an order $N_3 = p_1p_2p_3$, and a composite group of an order $N_4 = p_1p_2p_3p_4$ respectively. For evaluating the time complexity, we only take these two operations into account since the time cost of these two operations is much more than that of other operations like group add operation. The theoretical analysis of time complexity is shown in Table 2.

Table 2. Comparison with the previous schemes in time complexity

	KSW08 [5]	LOSTW10 [8]	OT12 (basic) [9]	Proposed
Setup	$O(n)\|T_{3G}\| + \|T_{3e}\|$	$O(n^2)\|T_G\| + \|T_e\|$	$O(n^2)\|T_G\| + \|T_e\|$	$O(n)\|T_{4G}\| + \|T_{4e}\|$
Encryption	$4n\|T_{3G}\|$	$O(n^2)\|T_G\|$	$O(n^2)\|T_G\|$	$2n\|T_{4G}\|$
Key generation	$4n\|T_{3G}\|$	$O(n^2)\|T_G\|$	$O(n^2)\|T_G\|$	$n\|T_{4G}\|$
Decryption	$2n\|T_{3e}\|$	$2n\|T_e\|$	$4n\|T_e\|$	$n\|T_{4e}\|$

Compared with the KSW08, our scheme needs fewer exponentiation operations in the encryption and key generation phases, and fewer pairing operations in the decryption phase. In contrast to LOSTW10 and OT12, our scheme only needs $O(n)$ exponentiation operations in the phase of setup, key generation and encryption. The efficiency of decryption in LOSTW10 and OT12 is better than ours since the time cost of pairing operations on a prime order group is much less than that on a composite order groups.

We denote the size of an element of G, $G3$ and $G4$ by $|G|$, $|G3|$ and $|G4|$, and that of G_T, G_{3T} and G_{4T} by $|G_T|$, $|G_{3T}|$ and $|G_{4T}|$ respectively. The comparison result of space complexity is shown in Table 3.

Table 3. Comparison with the previous schemes in space complexity

	KSW08 [5]	LOSTW10 [8]	OT12 (basic) [9]	Proposed
PK size	$O(n)\|G3\| + \|G_{3T}\|$	$O(n^2)\|G\| + \|G_T\|$	$O(n^2)\|G\| + \|G_T\|$	$O(n)\|G4\| + \|G_{4T}\|$
MSK size	$O(n)\|G3\|$	$O(n^2)\|G\|$	$O(n^2)\|G\|$	$O(n)\|G4\|$
SK size	$(2n+1)\|G3\|$	$(2n+3)\|G\|$	$(4n+2)\|G\|$	$(n+1)\|G4\|$
Cipher size	$(2n+1)\|G4\|$ $+\|G_{3T}\|$	$(2n+3)\|G\|$ $+\|G_T\|$	$(4n+2)\|G\|$ $+\|G_T\|$	$(n+1)\|G4\|$ $+\|G_{4T}\|$

Our scheme is better than KSW08 since our scheme needs fewer group elements in ciphertext and key. For LOSTW10 and OT12, the size of PK and MSK is linear with $O(n^2)$ due to using the technique called DPVS, which is less efficient than our scheme. For the storage cost of ciphertext and key, these schemes are constructed under the prime order group, which is more efficient than composite order group. However, they need more group elements than ours in each ciphertext and key. Thus, we need a detailed experiment (described in Section 5.2) to verify which scheme needs less space to store the ciphertext and key.

5.2. **Experimental results.** Table 4 shows the system configuration and the chosen elliptic curve of our experiments. Specifically, we implemented our construction in JAVA

Table 4. System configuration and elliptic curve

CPU	Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz
Memory	8GB
OS	Windows 7
Program Library	Java Pairing Based Cryptography library (JPBC)
Mathematical Parameters	
Elliptic Curve	$y^2 = x^3 + x$
Group Order	$2^a \pm 2^b \pm 1$ for random integers $0 < b < a$
The default unit is decimal	

with Java Pairing Based Cryptography library (JPBC) [32]. In our implementation, the bilinear map is instantiated as Type A pairing (base field size is 128-bit), which offers a level of security equivalent to 1024-bit DLOG [32]. Our experiment was run on Intel(R) Core(TM) i7-4790 CPU at 3.60GHz processor and 8GB memory size. Our experiment is based on the artificial plaintext indices with different number of documents (i.e., $D = 200; 400; 600; 800; 1000$) and different number of dimensions in a document (i.e., $n = 5; 10; 15; \dots; 45; 50$). In these indices, we set each dimension with a unique integer in a range $[-5000, 5000]$.

The experiment for each IPE scheme consists of the following steps:

1) By running its **setup** algorithm, PK and MSK can be generated;
2) Taking PK, test documents and its related vector as input, we preform **Enc** algorithm to obtain ciphertexts and then store these ciphertexts on our machine;
3) Using a random vector and MSK, we can create a key of this vector by using **KeyGen** algorithm; and
4) We apply the **Dec** algorithm to testing each ciphertext against the key.

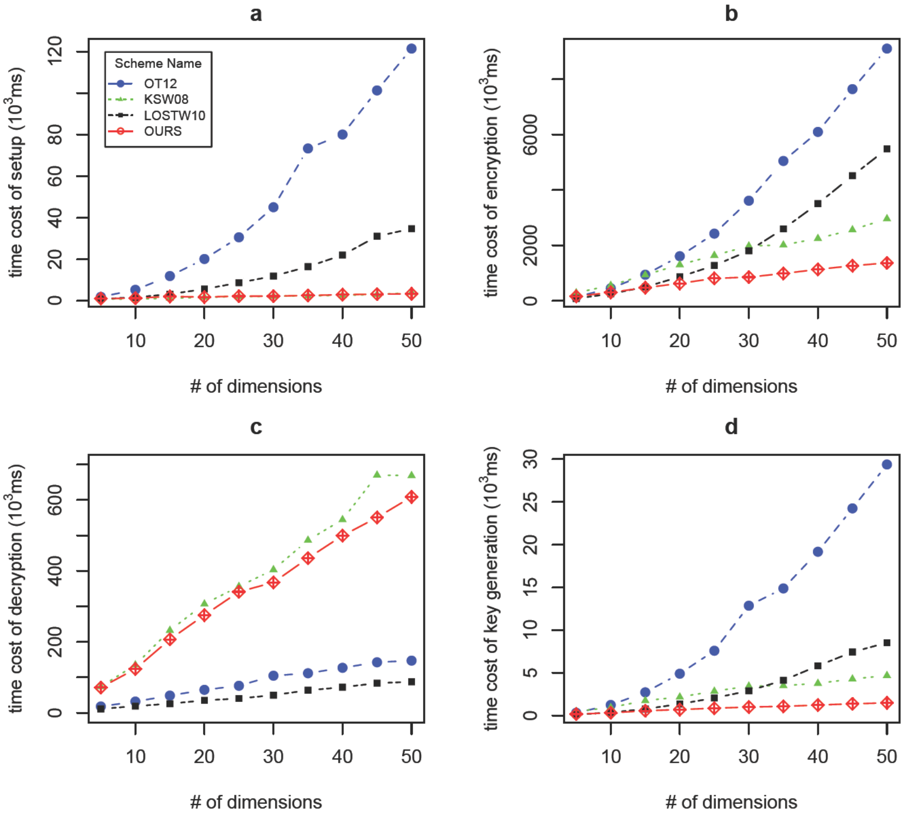

FIGURE 1. Impact of n on the time cost of setup (a), encryption (b), decryption (c) and key generation (d) in KSW08, LOSTW10, OT12 and the proposal. ($D = 600$, $n = \{5, 10, \dots, 45, 50\}$)

Time Overhead. For test documents with various dimension vectors ($n = 5, 10, \dots,$ 50), Figure 1 shows that:

1) Figure 1(a) shows that both the proposal and KSW08 have a better performance than other schemes in the Setup phase;
2) Figure 1(b) indicates that the encryption time in our scheme and KSW08 is linear with n while that in LOSTW10 and OT12 is related with n^2, and the time cost of our scheme is nearly a half of KSW08;

3) Figure 1(c) illustrates that the decryption time in all schemes is linear with n, and the time cost in the proposal is nearly four times more than that in the best scheme (LOSTW10); and
4) Figure 1(d) shows that, the time-consumption of our scheme is only one-third of that of KSW08 in the key generation phase, and more efficient than LOSTW10 and OT12.

Both our proposal and KSW08 are using composite order groups as a basic structure. However, our scheme needs fewer group elements in PK, MSK, ciphertext and key. That is to say, our proposal requires fewer exponentiation and pairing operations which indicates that our scheme is more efficient than KSW08 on the time complexity.

For the DPVS based schemes (LOSTW10 and OT12), we find that the time cost of these schemes increases with $O(n^2)$ in the setup, encryption and key generation phases, while the time consumption of these phases in our proposal is linear with $O(n)$. However, the DPVS-based schemes have better performance in decryption phase. The main reason is that the decryption algorithm in these DPVS-based schemes utilizes pairing operations over a prime order group, which is faster than the composite order group. Except this aspect, our proposal has better performance in the rest of phases.

For different sizes of document collection with the same dimension $n = 20$ ($D = 200, 400, 600, 800, 1000$), Figure 2 shows that:

1) Figure 2(a) shows that the time cost of encryption in all schemes is linear with $O(D)$, and the proposed scheme is the best one of all; and
2) Figure 2(b) indicates that the decryption time in all schemes is linear with $O(D)$, and LOSTW10 is the best one of all since this scheme is based on the prime order group.

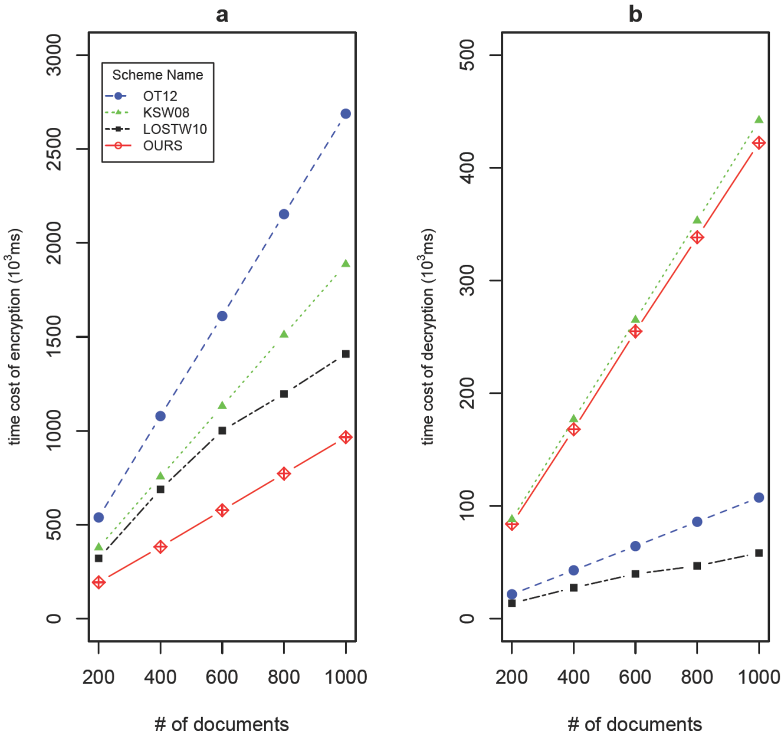

Figure 2. Impact of D on the time cost of encryption (a) and decryption (b) in KSW08, LOSTW10, OT12 and the proposal ($n = 20$, $D = \{200, 400, 600, 800, 1000\}$)

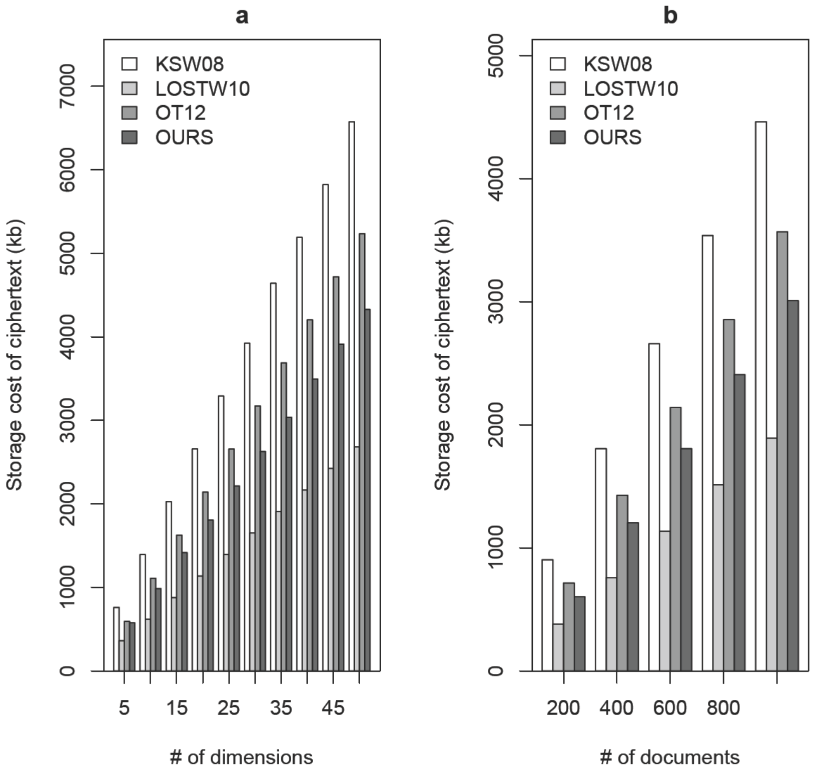

Figure 3. Impact of n (a) and D (b) on the storage cost of ciphertexts in KSW08, LOSTW10, OT12 and the proposal ($D = \{200, 400, 600, 800, 1000\}$, $n = \{5, 10, \ldots, 45, 50\}$)

Each document and key has its own attribute and predicate vector, respectively. Thus, both the encryption and decryption algorithm will preform D times when the document size is D. In addition, the time consumption in KeyGen and setup phases is independent with D since PK, MSK and keys are generated without using the information of documents.

Storage Overhead. From Figure 3(a), we can find that the storage cost of ciphertexts in all scheme is $O(n)$. That is because the number of elements in a ciphertext for all schemes is linear with n. Considering the fact that each document has its own ciphertext, as shown in Figure 3(b), we can find that the storage cost of ciphertexts in all schemes is linear with D. Because the number of elements in a ciphertext in our scheme is less than that in KSW08, our scheme is better than KSW08. Since LOSTW10 is based on the group of prime order, it needs less storage cost than the proposal. Although OT12 is also constructed on the prime order group, its ciphertext needs fourfold group elements than ours. Thus, our scheme is still slightly better than OT12.

From Table 5, we know that the storage cost of keys in all schemes is linear with n, and LOSTW10 is the best scheme on the space complexity of keys. From Table 6, we can argue that KSW08 and our scheme need less storage cost of PK and MSK. The reason is that the size of elements in the PK and MSK for the DPVS-based IPE schemes is quadratic to n while that for KSW08 and our scheme is linear to n. In summary, our scheme is better than any other schemes in the setup, encryption and key generation phases, but slightly worse than LOSTW10 and OT12 in the decryption phase. In the practical application, we can execute setup, encryption and key generation over the client terminals, while the decryption is executed over the server. Therefore, our proposal is more suitable for mobile networks, where client device usually has limited computation and storage resources.

Table 5. Comparison with the previous schemes in space cost of keys (100 keys)

Scheme \ Dimension size	10	20	30	40	50
KSW08	226	433	644	871	1085
LOSTW10	99	185	271	357	443
OT12	181	353	525	697	868
Proposal	151	288	424	569	698

Table 6. Comparison with the previous schemes in space cost of key pair (PK and MSK)

Scheme \ Dimension size	10	20	30	40	50
KSW08	3	5	7	9	12
LOSTW10	23	80	171	297	456
OT12	76	289	640	1128	1754
Proposal	4	6	9	12	15

6. **Conclusions.** In this paper, we proposed a new IPE scheme with better performance in encrypting and key generating under an adaptive security and weak attribute-hiding model. To reveal the efficiency of the proposed scheme, we compared it with the existing IPE schemes presented in [5, 8, 9] through theoretical analysis and experimental results.

According to the theoretical and experimental analysis presented in Section 4, we can argue that the proposed scheme is fit for the mobile cloud situation in which users use mobile device with less storage and limited computation capacity. Specifically, our scheme significantly reduces the time cost in the setup, encryption and key generation operations which are performed by clients. Moreover, the proposed scheme also needs less storage consuming of PK, MSK, ciphertexts and keys. We sacrifice a little security and decryption time as the slight compromise of the significant efficiency improvement. However, in the cloud setting, the decryption operation is performed by the cloud server which possesses strong computation and storage capacity; therefore, we argue that the trade-off is practical.

The future work is how to design an adaptively secure and fully attribute-hiding IPE scheme with less time and storage overhead in a prime order group.

Acknowledgments. The authors gratefully acknowledge the National Natural Science Foundation of China under Grant Nos. 61402393, 61601396, Shanghai Key Laboratory of Integrated Administration Technologies for Information Security (No. AGK201607) and Nanhu Scholars Program for Young Scholars of XYNU.

REFERENCES

[1] Z. Fu, K. Ren, J. Shu et al., Enabling personalized search over encrypted outsourced data with efficiency improvement, *IEEE Trans. Parallel and Distributed Systems*, vol.27, no.9, pp.2546-2559, 2016.

[2] P. S. Kumari and A. R. N. B. Kamal, An effective search of cloud data using ABC based cryptography and multiple keyword semantics, *International Journal of Innovative Computing, Information and Control*, vol.11, no.4, pp.1257-1267, 2015.

[3] Z. Fu, X. Sun, Q. Liu et al., Achieving efficient cloud search services: Multi-keyword ranked search over encrypted cloud data supporting parallel computing, *IEICE Trans. Commun.*, vol.98, no.1, pp.190-200, 2015.

[4] Z. Xia, X. Wang, X. Sun et al., A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data, *IEEE Trans. Parallel and Distributed Systems*, vol.27, no.2, pp.340-352, 2016.

[5] J. Katz, A. Sahai and B. Waters, Predicate encryption supporting disjunctions, polynomial equations, and inner products, *Advances in Cryptology – EUROCRYPT 2008, Lecture Notes in Computer Science*, vol.4965, pp.146-162, 2008.

[6] B. Waters, Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions, *Advances in Cryptology – CRYPTO 2009, Lecture Notes in Computer Science*, vol.5677, pp.619-636, 2009.

[7] A. Lewko and B. Waters, New techniques for dual system encryption and fully secure HIBE with short ciphertexts, *TCC 2010, Lecture Notes in Computer Science*, vol.5978, pp.455-479, 2010.

[8] A. Lewko, T. Okamoto, A. Sahai, K. Takashima and B. Waters, Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption, *Advances in Cryptology – EUROCRYPT 2010, Lecture Notes in Computer Science*, vol.6110, pp.62-91, 2010.

[9] T. Okamoto and K. Takashima, Adaptively attribute-hiding (hierarchical) inner product encryption, *EUROCRYPT 2012, the 31st International Conference on the Theory and Applications of Cryptographic Techniques*, vol.99, no.1, pp.591-608, 2012.

[10] D. Boneh and M. Franklin, Identity based encryption from the weil pairing, *CRYPTO 2001, Lecture Notes in Computer Science*, vol.2139, pp.213-229, 2001.

[11] C. Cocks, An identity based encryption scheme based on quadratic residues, *Proc. of the 8th IMA International Conference on Cryptography and Coding*, pp.26-28, 2001.

[12] M. Bellare, B. Waters and S. Yilek, Identity-based encryption secure against selective opening attack, *TCC 2011, Lecture Notes in Computer Science*, vol.6597, pp.235-252, 2011.

[13] C. Gentry, Practical identity-based encryption without random oracles, *Lecture Notes in Computer Science*, vol.4004, pp.445-464, 2006.

[14] J. Horwitz and B. Lynn, Toward hierarchical identity-based encryption, *Advances in Cryptology – EUROCRYPT 2002, Lecture Notes in Computer Science*, vol.2332, pp.466-481, 2002.

[15] C. Gentry and A. Silverberg, Hierarchical ID-based cryptography, *Advances in Cryptology – ASIACRYPT 2002, International Conference on the Theory and Application of Cryptology and Information Security*, Queenstown, New Zealand, pp.548-566, 2002.

[16] C. Gentry and S. Halevi, Hierarchical identity based encryption with polynomially many levels, *Proc. of the 6th Theory of Cryptography Conference on Theory of Cryptography*, pp.437-456, 2009.

[17] A. Sahai and B. Waters, Fuzzy identity-based encryption, *Advances in Cryptology – EUROCRYPT 2005*, pp.457-473, 2005.

[18] V. Goyal, O. Pandey, A. Sahai and B. Waters, Attribute-based encryption for fine-grained access control of encrypted data, *ACM Conference on Computer and Communications Security*, pp.89-98, 2006.

[19] X. Mao, J. Lai, Q. Mei et al., Generic and efficient constructions of attribute-based encryption with verifiable outsourced decryption, *IEEE Trans. Dependable and Secure Computing*, vol.13, no.5, pp.533-546, 2016.

[20] J. Han, W. Susilo, Y. Mu et al., Improving privacy and security in decentralized ciphertext-policy attribute-based encryption, *IEEE Trans. Information Forensics and Security*, vol.10, no.3, pp.665-678, 2015.

[21] Y. L. Lin, T. C. Wu and C. L. Hsu, Secure and efficient time-bound key assignment scheme for access control in hierarchical structure, *International Journal of Innovative Computing, Information and Control*, vol.6, no.2, pp.439-447, 2010.

[22] X. Boyen and B. Waters, Anonymous hierarchical identity-based encryption (without random oracles), *Advances in Cryptology – CRYPTO 2006, Lecture Notes in Computer Science*, vol.4117, pp.290-307, 2006.

[23] A. De Caro, V. Iovino and G. Persiano, Fully secure anonymous HIBE and secret-key anonymous IBE with short ciphertexts, *Proc. of the 4th International Conference on Pairing-Based Cryptography*, 2010.

[24] D. Boneh and B. Waters, Conjunctive subset and range queries on encrypted data, *TCC 2007, Lecture Notes in Computer Science*, vol.4392, pp.535-554, 2007.

[25] A. De Caro, V. Iovino and G. Persiano, Hidden vector encryption fully secure against unrestricted queries, *IACR Cryptology ePrint Archive*, 2011.

[26] K. Lee and D. H. Lee, Improved hidden vector encryption with short ciphertexts and tokens, *Designs, Codes and Cryptography*, doi:10.1007/s10623-010-9412-x, 2011.

[27] I. Kim, S. O. Hwang, J. H. Park et al., An efficient predicate encryption with constant pairing computations and minimum costs, *IEEE Trans. Computers*, vol.65, no.10, pp.2947-2958, 2016.

[28] M. Abdalla, R. Gay, M. Raykova et al., Multi-input inner-product functional encryption from pairings, *Advances in Cryptology – EUROCRYPT 2017*, 2017.

[29] P. Datta, T. Okamoto and J. Tomida, Full-hiding (unbounded) multi-input inner product functional encryption from the k-linear assumption, *IACR International Workshop on Public Key Cryptography*, Cham, pp.245-277, 2018.

[30] P. Datta, R. Dutta and S. Mukhopadhyay, Strongly full-hiding inner product encryption, *Theoretical Computer Science*, vol.667, pp.16-50, 2017.

[31] A. Joux, The Weil and Tate pairings as building blocks for public key cryptosystems (survey), *The 5th International Algorithmic Number Theory Symposium (ANTS-V)*, vol.2369, pp.20-32, 2002.

[32] A. De Caro and V. Iovino, jPBC: Java pairing based cryptography, *Proc. of the 16th IEEE Symposium on Computers and Communications*, http://gas.dia.unisa.it/projects/jpbc/, 2011.

Appendix.

Proof of Lemma 4.1.

Proof: Given g_1, g_3 and g_4, $\mathbb{B}$ can simulate $Game_{Real}$ with $\mathbb{A}$. With probability ϵ, $\mathbb{A}$ can generate the vectors $\vec{v}$ and $\vec{x}$ such that $\sum_{i=1}^{n} x_i v_i \bmod N \neq 0$ and $\sum_{i=1}^{n} x_i v_i \bmod p_2 = 0$. $\mathbb{B}$ uses these vectors to produce a nontrivial factor of N by computing $a = gcd\left(\sum_{i=1}^{n} x_i v_i, N\right)$. We set $b = \frac{N}{a}$. Notice that p_2 divides a and $N = ab = p_1 p_2 p_3 p_4$, we consider three cases:

1) Case 1: p_1 divides b.
2) Case 2: p_1 cannot divide b and p_4 can divide b.
3) Case 3: $a = p_1 p_2 p_4$ and $b = p_3$.

At least one of these cases must occur with probability $\geq \frac{\epsilon}{3}$. In Case (1), $\mathbb{B}$ will break Assumption 2.1. Given g_1, g_3, g_4 and T where $T \in G_{p_1}$ or $T \in G_{p_1 p_2}$, $\mathbb{B}$ can determine that p_1 divides b by verifying that ${g_1}^b$ is the identity element of G. Then $\mathbb{B}$ will compute T^b. If T^b is the identity element of G_T, then $T \in G_{p_1}$. Otherwise, $T \in G_{p_1 p_2}$.

Case (2) is the same as Case (1) except that $T \in G_{p_4}$ or $T \in G_{p_2 p_4}$. $\mathbb{B}$ can determine that p_1 cannot divide b and p_4 can divide b by verifying that ${g_1}^b$ is not the identity element of G and ${g_4}^b$ is the identity element of G. Then $\mathbb{B}$ will compute T^b. If T^b is the identity element of G_T, then $T \in G_{p_4}$. Otherwise, $T \in G_{p_2 p_4}$.

In Case (3), $\mathbb{B}$ will break Assumption 2.2. Given g_1, g_3, g_4, $D_1 D_2 \in G_{p_1 p_2}$, $B_2 B_3 \in G_{p_2 p_3}$ and T, $\mathbb{B}$ can determine that $a = p_1 p_2 p_4$ by verifying that $(D_1 D_2)^a$ is the identity element of G. Then $\mathbb{B}$ will compute $\hat{e}(T, (B_2 B_3)^b)$. If $\hat{e}(T, (B_2 B_3)^b)$ is the identity element of G_T, then $T \in G_{p_1 p_3}$. Otherwise, $T \in G_{p_1 p_2 p_3}$. □

Proof of Lemma 4.2.

Proof: Given g_1, g_3, g_4 and T, $\mathbb{B}$ can simulate $Game_0$ or $Game_{Restricted}$ with $\mathbb{A}$. To generate the public key, $\mathbb{B}$ chooses the random exponents α, a, b, c, u, a_i, b_i, α_i and $\beta_i \in Z_N$ and sets ${A_1}^{\beta_i} A_{4i} = {g_1}^{a\beta_i} {g_4}^{b_i}$, ${A_1}^{\alpha_i} B_{4i} = {g_1}^{a\alpha_i} {g_4}^{a_i}$, $B_1 = g_1^b$ and $U_1 U_4 = g_1^u g_4^c$ for each $i \in [1, n]$. Obviously, $U_1 = g_1^u$. B sends the public key $\left\{N, U_1 U_4, {A_1}^{\beta_i} A_{4i}, {A_1}^{\alpha_i} B_{4i}, g_4, \hat{e}(B_1, U_1)^\alpha\right\}$ to $\mathbb{A}$. Each time $\mathbb{A}$ asks $\mathbb{B}$ to provide a key for a vector $\vec{v}^{(j)} = \left(v_1^{(j)}, v_2^{(j)}, \ldots, v_n^{(j)}\right)$, where $j \in [1, q]$, $\mathbb{B}$ chooses $n+2$ random exponents r_j, c_{j1}, c_{j2}, …, c_{jn+1} and sets $K_{1i} = g_1^{ur_j \frac{v_i^{(j)}}{\beta_i}} g_3^{c_{ji}} = U_1^{r_j \frac{v_i^{(j)}}{\beta_i}} g_3^{c_{ji}}$ and $K_2 = {B_1}^\alpha {A_1}^{r_j \sum_{i=1}^{n} \frac{\alpha_i v_i^{(j)}}{\beta_i}} g_3^{c_{jn+1}}$.

After that, $\mathbb{A}$ sends $\mathbb{B}$ two messages $m^{(0)}$ and $m^{(1)}$ as well as two challenge vectors $\vec{x}^{(0)} = \left(x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}\right)$ and $\vec{x}^{(1)} = \left(x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)}\right)$. $\mathbb{B}$ randomly chooses $\beta \in \{0, 1\}$. Given C_{41}, C_{42}, …, C_{4n} and $C_4 \in G_{p_4}$ randomly (the random elements of G_{P_4} can be obtained by raising g_4 to the random exponents modulo N), $\mathbb{B}$ generates the ciphertext as follows:

$$C_0 = m^{(\beta)}\hat{e}(U_1, T)^{b\alpha}, \quad C_{1i} = T^{a\left(\beta_i x_i^{(\beta)} + \alpha_i\right)} C_{4i} \text{ and } C_2 = T^u C_4 \tag{17}$$

If $T \in G_{p_1p_2}$, then this ciphertext is a semi-functional ciphertext with $z_{ci} = a\Big(\beta_i x_i^{(\beta)} + \alpha_i\Big)$ and $z_c = u$. Since a, u, α_i and β_i are chosen randomly modulo N, it can be found that the value of z_{ci} and z_c module p_2 are unrelated with the value of a, u, α_i and β_i module p_1. If $T \in G_{p_1}$, this is a normal ciphertext. Therefore, if $\mathbb{A}$ can distinguish $Game_{Restricted}$ from $Game_0$ with advantage ϵ, $\mathbb{B}$ can use the output of $\mathbb{A}$ to break Assumption 2.1 with advantage ϵ. □

Proof of Lemma 4.3.

Proof: Given g_1, g_3, g_4, D_1D_2, B_2B_3 and T, $\mathbb{B}$ can simulate $Game_{k-1}$ or $Game_k$ with $\mathbb{A}$. Choosing the random exponents α, a, u, b, c, a_i, b_i, α_i and $\beta_i \in Z_N$, $\mathbb{B}$ sets ${A_1}^{\beta_i} A_{4i} = {g_1}^{a\beta_i}{g_4}^{b_i}$, ${A_1}^{\alpha_i} B_{4i} = {g_1}^{a\alpha_i}{g_4}^{a_i}$, $B_1 = g_1^b$, $U_1U_4 = g_1^u g_4^c$ and sends the public key $\left\{N, U_1U_4, {A_1}^{\beta_i}A_{4i}, {A_1}^{\alpha_i}B_{4i}, g_4, \hat{e}(B_1, U_1)^{\alpha}\right\}$ to $\mathbb{A}$. This sets $U_1 = g_1^u$. When A requests the j-th key for the vector $\vec{v}^{(j)} = \left(v_1^{(j)}, v_2^{(j)}, \ldots, v_n^{(j)}\right)$, $\mathbb{B}$ generates the normal key or the semi-function key for the vector $v^{(j)}$.

For $j < k$, $\mathbb{B}$ creates a semi-function key. Choosing the random exponents r_j, z_j and $t_{ji} \in Z_N$, $\mathbb{B}$ sets $K_{1i} = {g_1}^{\frac{ur_jv_i^{(j)}}{\beta_i}}(B_2B_3)^{t_{ji}}$ and $K_2 = {B_1}^{\alpha}{g_1}^{ar_j\sum_{i=1}^n \frac{\alpha_i v_i^{(j)}}{\beta_i}}(B_2B_3)^{z_j}$. This is a properly distributed semi-functional key with $g_2^{\xi z_k} = {B_2}^{z_j}$ and $g_2^{\xi z_{ki}} = {B_2}^{t_{ji}}$.

For $j > k$, $\mathbb{B}$ generates the normal keys by choosing the random exponents r_j, w_j and $t_{ji} \in Z_N$ and setting $K_{1i} = {g_1}^{\frac{ur_jv_i^{(j)}}{\beta_i}}(g_3)^{t_{ji}}$ and $K_2 = {B_1}^{\alpha}{g_1}^{ar_j\sum_{i=1}^n \frac{\alpha_i v_i^{(j)}}{\beta_i}}(g_3)^{w_j}$.

To create the k-th requested key, choosing $n+1$ random exponents $w_{k1}, w_{k2}, \ldots, w_{kn+1}$, $\mathbb{B}$ sets $K_{1i} = T^{\frac{uv_i^{(k)}}{\beta_i}} g_3^{w_{ki}}$ and $K_2 = {B_1}^{\alpha}T^{a\sum_{i=1}^n \frac{\alpha_i v_i^{(k)}}{\beta_i}}(g_3)^{w_{kn+1}}$. It can be found that $z_{ki} = \frac{uv_i^{(k)}}{\beta_i}$ and $z_k = a\sum_{i=1}^n \frac{\alpha_i v_i^{(k)}}{\beta_i}$.

After the key request phase, $\mathbb{A}$ sends $\mathbb{B}$ two messages $m^{(0)}$ and $m^{(1)}$, and two challenge vectors $\vec{x}^{(0)}$ and $\vec{x}^{(1)}$. $\mathbb{B}$ randomly chooses $\beta \in \{0, 1\}$ and generates the semi-function ciphertext. For the challenge message $m^{(\beta)}$ and the challenge vector $\vec{x}^{(\beta)} = \left(x_1^{(\beta)}, x_2^{(\beta)}, \ldots, x_n^{(\beta)}\right)$, $\mathbb{B}$ generates $n+1$ random elements $C_{41}, C_{42}, \ldots, C_{4n}, C_4 \in G_{P4}$ by taking g_4 and raising it to a random exponents module N and sets $C_0 = m^{(\beta)}\hat{e}(D_1D_2, U_1)^{b\alpha}$, $C_{1i} = (D_1D_2)^{a\left(\beta_i x_i^{(\beta)} + \alpha_i\right)}C_{4i}$ and $C_2 = (D_1D_2)^u C_4$.

Note that this sets $z_c = u$ and $z_{ci} = a\left(\beta_i x_i^{(\beta)} + \alpha_i\right)$. Since a, u, α_i and β_i are chosen randomly modulo N, we can find that the value of z_{ci} and z_c module p_2 are unrelated with the value of a, u, α_i and β_i module p_1. Suppose that $\vec{x}^{(\beta)} \cdot v^{(k)} = \sum_{i=1}^n x_i^{(\beta)} v_i^{(k)} = 0 \bmod p_2$, It be found that $z_c z_k = \sum_{i=1}^n z_{ci} z_{ki} \bmod p_2$. It means that if $\vec{x}^{(\beta)} \cdot v^{(k)} = 0 \bmod p_2$, $\mathbb{A}$ has made an invalid key request. This is where we use our additional modular restriction. Therefore, according to the $Game_{Restricted}$, as long as $\vec{x}^{(\beta)} \cdot v^{(k)} \neq 0 \bmod p_2$, z_k, z_c, z_{ki} and z_{ci} are randomly distributed to $\mathbb{A}$.

Besides, we observe that, if $\mathbb{B}$ attempts to test whether the k-th key is semi-functional by creating a semi-functional ciphertext for the vector $\vec{x}^{(k)}$ such that $\vec{x}^{(k)}\vec{v}^{(k)} = 0$ and trying to decrypt it, $\mathbb{B}$ can find that the decryption can still work whether the k-th key is semi-functional or not since $z_c z_k = \sum_{i=1}^n z_{ci} z_{ki}$.

Therefore, we can conclude that, if $T \in G_{p_1p_3}$, $\mathbb{B}$ has properly simulated $Game_{k-1}$. If $T \in G_{p_1p_2p_3}$, $\mathbb{B}$ has properly simulated $Game_k$. So, if $\mathbb{A}$ can distinguish $Game_{k-1}$

from $Game_k$ with advantage ϵ, $\mathbb{B}$ can use the output of $\mathbb{A}$ to break Assumption 2.2 with advantage ϵ. □

Proof of Lemma 4.4.

Proof: Given g_1, g_2, g_3, g_4, $g_1^\alpha A_2$, $g_1^s B_2$, g_2^r, A_2^r and T, $\mathbb{B}$ chooses the random exponents a, b, c, u, a_i, b_i, α_i and $\beta_i \in Z_N$ and sets ${A_1}^{\beta_i} A_{4i} = {g_1}^{a\beta_i} {g_4}^{b_i}$, ${A_1}^{\alpha_i} B_{4i} = {g_1}^{a\alpha_i} {g_4}^{a_i}$, $B_1 = g_1^b$, $U_1 U_4 = g_1^u g_4^c$ and $\hat{e}(B_1, U_1)^\alpha = \hat{e}\left(g_1^\alpha A_2, U_1\right)^b$. Obviously, $U_1 = g_1^u$. Then $\mathbb{B}$ sends the public key $\left\{N, U_1 U_4, {A_1}^{\beta_i} A_{4i}, {A_1}^{\alpha_i} B_{4i}, g_4, \hat{e}(B_1, U_1)^\alpha\right\}$ to $\mathbb{A}$. When $\mathbb{A}$ requests a key for a vector $\vec{v}^{(j)} = \left(v_1^{(j)}, v_2^{(j)}, \ldots, v_n^{(j)}\right)$, $\mathbb{B}$ generates a semi-functional key. It does this by choosing the random exponents r_j, c_{j1}, c_{j2}, …, c_{jn+1}, t_{j1}, t_{j2}, …, $t_{jn+1} \in Z_N$ and setting $K_{1i} = {U_1}^{r_j \frac{v_i^{(j)}}{\beta_i}} {A_2}^{rt_{ji}} {g_3}^{c_{ji}}$ and $K_2 = \left({g_1}^\alpha A_2\right)^b {A_1}^{r_j \sum_{i=1}^n \frac{\alpha_i v_i^{(j)}}{\beta_i}} {A_2}^{rt_{jn+1}} {g_3}^{c_{jn+1}} = {B_1}^\alpha A_2^b {A_1}^{r_j \sum_{i=1}^n \frac{\alpha_i v_i^{(j)}}{\beta_i}} {A_2}^{rt_{jn+1}} {g_3}^{c_{jn+1}}$.

$\mathbb{A}$ sends $\mathbb{B}$ two messages $m^{(0)}$ and $m^{(1)}$, and two challenge vectors $\vec{x}^{(0)}$ and $\vec{x}^{(1)}$. $\mathbb{B}$ randomly chooses $\beta \in \{0,1\}$, creates $n+1$ random elements $C_{41}, C_{42}, \ldots, C_{4n}, C_4 \in G_{P4}$ by taking g_4 and raising it to a random exponents module N and generates the semi-function ciphertext as follows:

$$C_0 = m^{(\beta)} T^{ub}, \quad C_{1i} = (g_1^s B_2)^{a\left(\beta_i x_i^{(\beta)} + \alpha_i\right)} C_{4i}, \quad C_2 = (g_1^s B_2)^u C_4, \tag{18}$$

where $i \in [1, n]$.

This sets $z_{ci} = a\left(\beta_i x_i^{(\beta)} + \alpha_i\right)$ and $z_c = u$. Since a, u, α_i and β_i are chosen randomly modulo N, the value of z_{ci} and z_c module p_2 are unrelated with the value of a, u, α_i and β_i module p_1. If $T = \hat{e}(g_1, g_1)^{\alpha s}$, this is a properly distributed semi-functional ciphertext with message $m^{(\beta)}$ and $\mathbb{B}$ has properly simulated $Game_q$. If T is a random element of G_T, the ciphertext is a semi-functional ciphertext with a random message and $\mathbb{B}$ has properly simulated $Game_{Final_0}$.

Therefore, if $\mathbb{A}$ can distinguish the $Game_q$ from $Game_{Final_0}$ with advantage ϵ, $\mathbb{B}$ can use the output of $\mathbb{A}A$ to break Assumption 2.3 with advantage ϵ. □

Proof of Lemma 4.5.

Proof: Given g_2, g_3, g_4, $A_1 A_4$, $E_1 E_2$ and T, $\mathbb{B}$ chooses the random exponents b, c, u, α, a_i, b_i, α_i, $\beta_i \in Z_N$, generates ${A_1}^{\beta_i} A_{4i} = (A_1 A_4)^{\beta_i} g_4^{a_i}$, ${A_1}^{\alpha_i} B_{4i} = (A_1 A_4)^{\alpha_i} g_4^{b_i}$, $U_1 U_4 = (A_1 A_4)^u g_4^c$ and $\hat{e}(B_1, U_1)^\alpha = \hat{e}\left((E_1 E_2)^b, U_1 U_4\right)^\alpha$. This implicitly sets $U_1 = A_1^u$ and $B_1 = E_1^b$. Then, $\mathbb{B}$ sends the public key $\left\{N, U_1 U_4, {A_1}^{\beta_i} A_{4i}, {A_1}^{\alpha_i} B_{4i}, g_4, \hat{e}(B_1, U_1)^\alpha\right\}$ to $\mathbb{A}$. Each time $\mathbb{B}$ is asked to provide a key for a vector $\vec{v}^{(j)} = \left(v_1^{(j)}, v_2^{(j)}, \ldots, v_n^{(j)}\right)$, suppose that $E_1 = A_1^{e_1}$, where $e_1 \in Z_n$, then, by choosing the random exponents r_j, w_{j1}, w_{j2}, …, w_{jn+1}, z_{j1}, z_{j2}, …, $z_{jn+1} \in Z_N$, $\mathbb{B}$ creates a semi-functional key as follows:

$$K_{1i} = (E_1 E_2)^{ur_j \frac{v_i^{(j)}}{\beta_i}} g_2^{w_{ji}} g_3^{z_{ji}} = U_1^{e_1 r_j \frac{v_i^{(j)}}{\beta_i}} E_2^{ur_j \frac{v_i^{(j)}}{\beta_i}} g_2^{w_{ji}} g_3^{z_{ji}} \tag{19}$$

$$\begin{aligned} K_2 &= (E_1 E_2)^{b\alpha} (E_1 E_2)^{r_j \sum_{i=1}^n \frac{\alpha_i v_i^{(j)}}{\beta_i}} g_2^{w_{jn+1}} g_3^{z_{jn+1}} \\ &= B_1^\alpha A_1^{e_1 r_j \sum_{i=1}^n \frac{\alpha_i v_i^{(j)}}{\beta_i}} E_2^{b\alpha + r_j \sum_{i=1}^n \frac{\alpha_i v_i^{(j)}}{\beta_i}} g_2^{w_{jn+1}} g_3^{z_{jn+1}} \end{aligned} \tag{20}$$

At some point, $\mathbb{A}$ sends $\mathbb{B}$ two messages $m^{(0)}$ and $m^{(1)}$, and two challenge vectors $\vec{x}^{(0)}$ and $\vec{x}^{(1)}$. Randomly choosing $\beta \in \{0,1\}$, a random element $R \in G_T$ and $w_i, y_i, z_i, r_2, r_4, s \in Z_N$, $\mathbb{B}$ creates the challenge ciphertext as follows:

$$C_0 = R \tag{21}$$

When $i \leq k-1$, we have

$$C_{1i} = (A_1 A_4)^{w_i} g_2^{y_i} g_4^{z_i} = A_1^{w_i} C_{24}^i \tag{22}$$

where $C_{24}^i = g_2^{y_i} g_4^{z_i} A_4^{w_i}$.

When $i = k$,

$$C_{1k} = \left({A_1}^{\beta_k} A_{4k}\right)^{sx_k^{(\beta)}} \times ({A_1}^{\alpha_k} B_{4k})^s \times g_2^{y_k} g_4^{z_k} \times T = A_1^{s\left(\beta_k x_k^{(\beta)} + \alpha_k\right)} C_{24}^k T \tag{23}$$

where $C_{24}^k = A_{4k}^{sx_k^{(\beta)}} B_{4k}^s g_2^{y_k} g_4^{z_k}$.

When $i \geq k+1$,

$$C_{1i} = \left({A_1}^{\beta_i} A_{4i}\right)^{sx_i^{(\beta)}} \times ({A_1}^{\alpha_i} B_{4i})^s \times g_2^{y_i} g_4^{z_i} = A_1^{s\left(\beta_i x_i^{(\beta)} + \alpha_i\right)} C_{24}^i \tag{24}$$

where $C_{24}^i = A_{4i}^{sx_i^{(\beta)}} B_{4i}^s g_2^{y_i} g_4^{z_i}$.

$$C_2 = (U_1 U_4)^s \times g_2^{r_2} g_4^{r_4} = U_1^s R_{24} \tag{25}$$

where $R_{24} = g_2^{r_2} g_4^{r_4} U_4^s$.

Since $A_1 A_4$ is chosen randomly in G_1 and α_i, β_i, y_i, u, s are chosen randomly in Z_N, the value of z_{ci} and z_c module p_2 are unrelated with the value of s, u, α_i and β_i module p_1.

If $T \in G_{p_2 p_4}$, then this is a properly distributed semi-functional ciphertext with a random message and a challenge vector where its first $k-1$ elements are random and the rest elements are normal. In this case, $\mathbb{B}$ has properly simulated $Game_{Final_{1,k-1}}$. If $T \in G_{p_1 p_2 p_4}$, this is a properly distributed semi-functional ciphertext with a random message and a challenge vector where its first k elements are random and the rest elements are normal. In this case, $\mathbb{B}$ has properly simulated $Game_{Final_{1,k}}$.

Therefore, if $\mathbb{A}$ can distinguish $Game_{Final_{1,k-1}}$ from $Game_{Final_{1,k}}$ with advantage ϵ, $\mathbb{B}$ can use the output of $\mathbb{A}$ to break Assumption 2.4 with advantage ϵ. □