Improving the Message-ciphertext Rate of Lewko's Fully Secure IBE Scheme

Dingding Jia[1], Bao Li[1], Yamin Liu[1], and Qixiang Mei[2]

{ddjia,lb,ymliu}@is.ac.cn

Abstract. In Eurocrypt 2012, Lewko presented a fully secure IBE scheme in the prime order setting based on the decisional linear assumption. We note that some random factor involved in the ciphertext can further be used to hide yet another message , and get a new fully secure IBE scheme with better message-ciphertext rate. Similar to Lewko's scheme, we use dual pairing vector space in prime order bilinear groups to simulate the canceling and parameter hiding properties of composite order settings. The security of our scheme is based on the subspace assumption, which can be reduced to the decisional linear assumption. We employ the dual system encryption technique in our security proof.

Key words: DLIN assumption, fully secure IBE, canceling, parameter hiding, dual system encryption

1 Introduction

In 1984, Shamir [16] introduced the notion of Identity-Based Encryption (IBE). An IBE is a public key encryption scheme in which the public key can be set to any string representing one's identity. A trusted authority holds a master secret key which allows it to create secret keys for any identity. There are two kinds of security requirements for IBE schemes: a weaker one called selective-ID security in which the adversary selects an ID priori to other moves and attacks the fixed ID; and a stronger one called fully security in which the adversary adaptively selects the ID to be attacked during the security game. IBE schemes are first realized in the random oracle model, by Boneh and Franklin [3] using bilinear groups and Cocks [7] under quadratic residue assumption. Later, realization in the standard model was proposed by Boneh and Boyen [2] and Canetti, Halevi and Katz [6], but only selective-ID security was achieved in [2, 6].

The first fully secure IBE scheme that has a tight reduction in the standard model was proposed by Waters in 2009 [17], in which a new proof

technique called dual system encryption was used. The IBE scheme of [17] was constructed with bilinear maps in the prime order setting, and its security was based on the decisional linear (DLIN) assumption and bilinear decisional Diffie-Hellman (BDDH) assumption. However, the scheme was bothered by too long parameters and complicated structure. In [15] Ramannal, et al. gave a simplification of Waters' scheme in asymmetric bilinear groups, but based on assumptions which are not so standard.

Shortly later another fully secure IBE scheme was given by Lewko and Waters in 2010 [12] in the composite order setting. Their scheme has simple structure resembles that of [2]. In the security proof they used two properties of composite order groups: one is called "canceling", that is, for any $g_1 \in G_{p_1}, g_2 \in G_{p_2},\ e(g_1, g_2) = 1$; the other is called "parameter hiding", which means g_1^a information-theoretically hides $a\ mod\ p_2$.

Since the computation of bilinear map in composite order groups is less efficient, much effort has been contributed to finding transformations to prime order settings. In 2010, Freeman [8] provided a generic method for transforming schemes in composite order settings [4, 9, 11] to prime order settings, but the method can not be applied to some schemes. Lewko [10] observed that the method of [8] perfectly simulated the "canceling" property, yet was not a useful approach to achieve the "parameter hiding" property. Lewko [10] used dual pairing vector space which was proposed by Okamoto and Takashima [13, 14] to simulate both properties in the prime order setting, and got a fully secure IBE scheme akin to the one in [2]. Specially, to achieve the canceling property, a pair of dual orthonormal bases $(\mathbb{B}, \mathbb{B}^*)$ was used in [10]; to achieve the parameter hiding property, for a matrix A to be hidden, a pair of random dual orthonormal bases $(\mathbb{B}, \mathbb{B}^*)$ was chosen, then A was embedded in $(\mathbb{B}, \mathbb{B}^*)$, and a new pair of dual orthonormal bases $(\mathbb{B}_A, \mathbb{B}_A^*)$ was generated, which looks random to the adversary who does not know $(\mathbb{B}, \mathbb{B}^*)$. The scheme in [10] has simple structure and its security is based on the DLIN assumption.

In the IBE sheme in [10], both ciphertext and secret key employ two parameters, s_1, s_2 and r_1, r_2. The two exponents play almost the same roles, except that in the ciphertext, the element hiding the message only uses s_1. We find that both parameters could be used to hide messages, thus more messages could be encrypted without adding too many elements to the ciphertext.

Our Result. In this paper, we improve the IBE scheme presented in [10] by using both parameters to hide messages in the ciphertext, hence

get an IBE scheme that can encrypt two elements in the target group simultaneously. As in [10], we use dual orthonormal bases of dimension 6. Compared to Lewko's scheme, we only add one element from the target group to the public parameters and ciphertexts, double the length of the decryption key. In table 1 we give a comparison of Lewko's scheme and ours. The columns #PP, #Msk,#msg, #cpr, #key provide the number of group elements in the public parameters, the master secret key, messages, ciphertexts and decryption keys. Encryption efficiency counts the number of scalar multiplications in G for every group element while decryption efficiency counts the number of pairings that are required. Key generation efficiency is given by the number of scalar multiplications in G. We can see that we encrypt double messages with little extra cost, thus our scheme has better message-ciphertext rate. Similar to Lewko's paper, we use dual pairing vector space in prime order bilinear groups to realize the canceling and parameter hiding properties.

Scheme	#PP	#Msk	#msg	#cpr	#key	enc eff	dec eff	keygen eff	msg-cpr rate
Lewko's	25	30	1	7	6	25	6	6	1:7
Ours	26	36	2	8	12	13	6	12	2:8

Table 1. A comparison of Lewko's IBE scheme and ours

In the security proof we use the dual system encryption technique. Firstly we change the challenge ciphertext to be semi-functional. Secondly answers to key extraction queries are changed to be semi-functional one by one. Here we change every key in two steps: temporary semi-functional first, then semi-functional. Finally we change the challenge ciphertext to a semi-functional encryption of a random message. We argue that any PPT adversary can not tell the difference between two adjacent games.

Moreover, in the last game the challenge ciphertext is independent of the identity, so our scheme is anonymous. The IBE scheme in Lewko's paper is also anonymous for the same reason. Anonymous IBE [5] is a useful component to construct public key encryption with keyword search (PEKS) schemes [1].

The rest of our paper is organized as follows: in section 2 we give the formal definition of IBE and the security definition; in section 3 we give the complexity assumptions; in section 4 we describe our construction and prove its security; section 5 is the conclusion of the whole paper.

2 Definitions

2.1 IBE

Definition 1 (IBE). *An Identity-Based Encryption scheme (IBE) [16] is a tuple of four probabilistic polynomial time (PPT) algorithms: (**Setup, Keygen, Encrypt, Decrypt.**)*

$Setup(1^\lambda)$**:** *take as input the security parameter λ and output public parameters PP and the master secret key Msk.*
$Keygen(Msk, ID)$**:** *take as input the master secret key Msk, identity ID and output a private key Sk_{ID}.*
$Encrypt(PP, M, ID)$**:** *take as input the public paramerters PP, message M and identity ID and output a ciphertext C.*
$Decrypt(C, Sk_{ID})$**:** *take as input the ciphertext C and secret key Sk_{ID} and output the message M.*

For correctness, we require that all properly generated ciphertexts can be decrypted correctly.

2.2 Security Definition

Here we give the fully security definition of IBE. The security of an IBE scheme is defined using the following game between an adversary $\mathcal{A}$ and a challenger.

Setup: The challenger runs the *Setup* algorithm, gives the public parameters PP to the adversary $\mathcal{A}$ and keeps the master secret key Msk to itself.
Phase 1: $\mathcal{A}$ adaptively issues identity queries ID, the challenger responds with Sk_{ID} by calling the $Keygen$ algorithm.
Challenge: $\mathcal{A}$ gives two messages and a challenge identity (M_0, M_1, ID^*) to the challenger. The challenge identity should never be queried in phase 1. The challenger picks a random bit b and responds with $Encrypt(PP, M_b, ID^*)$.
Phase 2: $\mathcal{A}$ adaptively issues additional queries as in Phase 1, with the restriction that ID^* is never allowed to be queried.
Guess: $\mathcal{A}$ outputs a guess b' of b.

The advantage of $\mathcal{A}$ is defined as $Adv_{\mathcal{A}}^{IBE} = \left| Pr[b' = b] - \frac{1}{2} \right|$.

Definition 2 (Fully Security). *An IBE scheme is fully secure if for all PPT adversary $\mathcal{A}$, $Adv_{\mathcal{A}}^{IBE}$ is negligible in λ.*

3 Complexity Assumptions

In this section we introduce the complexity assumptions that will be used in our proof. As in [10], we use dual pairing vector space to achieve the canceling and parameter hiding properties in the prime order setting.

3.1 Prime Order Symmetric Bilinear Maps

Let G, G_T be cyclic groups of prime order p, with a bilinear map $e : G \times G \to G_T$ satisfying the following properties:

- (Bilinear) $\forall u_1, u_2 \in G, \forall a, b \in \mathbb{Z}_p, e(u_1^a, u_2^b) = e(u_1, u_2)^{ab}$.
- (Non-degenerate) $\exists g \in G$ such that $e(g, g)$ has order p in G_T.

We assume that the group operations in G and $\mathbb{G}_T$ as well as the bilinear map e can be efficiently computed.

Vector computation rules are defined as follows:

- For $\mathbf{v} = (v_1, v_2, ..., v_n) \in \mathbb{Z}_p^n$ and $g \in G, g^{\mathbf{v}} := (g^{v_1}, g^{v_2}, ..., g^{v_n})$.
- For any $a \in \mathbb{Z}_p$ and $\mathbf{v}, \mathbf{w} \in \mathbb{Z}_p^n$,

$$g^{a\mathbf{v}} := (g^{av_1}, ..., g^{av_n}), g^{\mathbf{v}+\mathbf{w}} := (g^{v_1+w_1}, ..., g^{v_n+w_n}).$$

- e_n is used to denote the product of the component-wise pairings:

$$e_n(g^{\mathbf{v}}, g^{\mathbf{w}}) := \prod_{i=1}^{n} e(g^{v_i}, g^{w_i}) = e(g, g)^{\mathbf{v} \cdot \mathbf{w}}.$$

Dual Pairing Vector Spaces. Next let us review the concept of dual pairing vector spaces from [13, 14, 10]. For a dimension n, we say two random

chosen bases $\mathbb{B} := (\mathbf{b}_1, ..., \mathbf{b}_n), \mathbb{B}^* := (\mathbf{b}_1^*, ..., \mathbf{b}_n^*)$ are dual orthonormal if for $i \neq j$,

$$\mathbf{b}_i \cdot \mathbf{b}_j^* = 0 (mod\ p)$$

and $\mathbf{b}_i \cdot \mathbf{b}_i^* = \psi$ for all i, where ψ is a uniformly random element in $\mathbb{Z}_p$. In the following we let $Dual(\mathbb{Z}_p^n)$ be the set of dual orthonormal bases and let $(\mathbb{B}, \mathbb{B}^*) \xleftarrow{R} Dual(\mathbb{Z}_p^n)$ denote choosing random dual orthonormal bases $\mathbb{B}$ and $\mathbb{B}^*$ from $\mathbb{Z}_p^n$.

Canceling Property. For a generator $g \in G$, we note that $e_n(g^{\mathbf{b}_i}, g^{\mathbf{b}_j^*}) = 1$ whenever $i \neq j$. We call this property as "canceling" and it will play an important role in our scheme.

Parameter Hiding Property. Next we will introduce the other property called "parameter hiding". Generally speaking, one can apply an invertible matrix A to a random pair of dual orthonormal bases $(\mathbb{B}, \mathbb{B}^*) \xleftarrow{R} Dual(\mathbb{Z}_p^n)$, and get a new pair of dual orthonormal bases which is randomly distributed for adversaries who do not know $(\mathbb{B}, \mathbb{B}^*)$. Hence the newly generated bases information-theoretically hide the matrix A. Next we describe how the new pair of dual orthonormal bases is generated.

For $(\mathbb{B}, \mathbb{B}^*) \xleftarrow{R} Dual(\mathbb{Z}_p^n)$, let $m < n$ be a fixed number, and A be an invertible $m \times m$ matrix. We use S_m to denote a subset of $[n]$ satisfying $|S_m| = m$, let $\mathbb{B}_m$ be an $n \times m$ matrix consists of $\mathbf{b}_i$ for $i \in S_m$. Associated with S_m we define $\mathbb{B}_A$ as follows: for $i \notin S_m$, keep $\mathbf{b}_i$ unchanged; for $i \in S_m$, swap $\mathbf{b}_i$ for the corresponding column in $\mathbb{B}_m A$. We get $\mathbb{B}_A^*$ in a similar way except that for $i \in S_m$, we swap $\mathbf{b}_i^*$ for the corresponding column in $\mathbb{B}_m^*(A^{-1})^T$. From Lemma 1 in [10] we get that $(\mathbb{B}_A, \mathbb{B}_A^*)$ is distributed as random dual orthonormal bases as long as $(\mathbb{B}, \mathbb{B}^*)$ is randomly chosen. Especially, the pair $(\mathbb{B}_A, \mathbb{B}_A^*)$ information-theoretically hides A.

3.2 Complexity Assumptions

Decisional Linear Assumption (DLIN). To formally define our assumption, we let $\mathcal{G}$ denote a group generation algorithm, which takes in a security parameter λ and outputs a symmetric bilinear map e together with G, G_T of order p.

Let $\mathcal{G}$ be a group generator, run $\mathcal{G}(1^\lambda)$ to get (p, G, G_T, e), and randomly choose $g, f, v \in G, c_1, c_2, w \in \mathbb{Z}_p$, $T_0 = g^{c_1+c_2}, T_1 = g^w$. The advan-

tage of $\mathcal{A}$ is defined as

$$Adv_{\mathcal{A}}^{DLIN} = \Big|Pr[\mathcal{A}(g, f, v, f^{c_1}, v^{c_2}, T_1) = 1] - Pr[\mathcal{A}(g, f, v, f^{c_1}, v^{c_2}, T_0) = 1]\Big|.$$

Definition 3 (DLIN). *We say that $\mathcal{G}$ satisfies DLIN assumption if for all PPT algorithm $\mathcal{A}$, $Adv_{\mathcal{A}}^{DLIN}$ is negligible in λ.*

Next we describe the subspace assumption in [10], here we require that $k \leq \frac{n}{3}$.

Subspace Assumption. Let $\mathcal{G}$ be a group generator algorithm as above, run $\mathcal{G}(1^\lambda)$ to get (p, G, G_T, e), and randomly choose

$$(\mathbb{B}, \mathbb{B}^*) \xleftarrow{R} Dual(\mathbb{Z}_p^n), g \xleftarrow{R} G,$$

$$\eta, \beta, \mu_1, \mu_2, \mu_3, \tau_1, \tau_2, \tau_3 \xleftarrow{R} \mathbb{Z}_p,$$

$$U_1 := g^{\mu_1 \mathbf{b}_1 + \mu_2 \mathbf{b}_{k+1} + \mu_3 \mathbf{b}_{2k+1}}, ..., U_k := g^{\mu_1 \mathbf{b}_k + \mu_2 \mathbf{b}_{2k} + \mu_3 \mathbf{b}_{3k}},$$

$$V_1 := g^{\tau_1 \eta \mathbf{b}_1^* + \tau_2 \beta \mathbf{b}_{k+1}^*}, ..., V_k := g^{\tau_1 \eta \mathbf{b}_k^* + \tau_2 \beta \mathbf{b}_{2k}^*},$$

$$W_1 := g^{\tau_1 \eta \mathbf{b}_1^* + \tau_2 \beta \mathbf{b}_{k+1}^* + \tau_3 \mathbf{b}_{2k+1}^*}, ..., W_k := g^{\tau_1 \eta \mathbf{b}_k^* + \tau_2 \beta \mathbf{b}_{2k}^* + \tau_3 \mathbf{b}_{3k}^*},$$

$$D := (g^{\mathbf{b}_1}, g^{\mathbf{b}_2}, ..., g^{\mathbf{b}_{2k}}, g^{\mathbf{b}_{3k+1}}, ..., g^{\mathbf{b}_n}, g^{\eta \mathbf{b}_1^*}, ..., g^{\eta \mathbf{b}_k^*},$$
$$g^{\beta \mathbf{b}_{k+1}^*}, ..., g^{\beta \mathbf{b}_{2k}^*}, g^{\mathbf{b}_{2k+1}^*}, ..., g^{\mathbf{b}_n^*}, U_1, ..., U_k, \mu_3)$$

The advantage of $\mathcal{A}$ is defined as

$$Adv_{\mathcal{A}}^{SA} = \Big|Pr[\mathcal{A}(D, V_1, ..., V_k) = 1] - Pr[\mathcal{A}(D, W_1, ..., W_k) = 1]\Big|.$$

In this paper we use subspace assumption with $n = 6$ and $k = 2$ or $k = 1$.

Definition 4. *We say that $\mathcal{G}$ satisfies the subspace assumption if for all PPT algorithm $\mathcal{A}$, $Adv_{\mathcal{A}}^{SA}$ is negligible in λ.*

It was shown in [10] that the subspace assumption can be reduced to DLIN assumption.

Lemma 1. *[10] If there is an adversary $\mathcal{A}$ that can break the subspace assumption with probability ϵ, then we can build an algorithm $\mathcal{B}$ having the same advantage in solving the DLIN problem.*

4 An IBE Scheme with Better Message-ciphertext Rate

4.1 Our Construction

In this section we describe our construction of IBE scheme. The structure of our scheme is similar to that in [10], however, by using both random parameters in the ciphertext to hide messages, we can get an IBE scheme that can encrypt two elements from G_T. Also we note that in Lewko's scheme they used θ, σ along with $\mathbf{d}_i^*, i = 1, ..., 4$, but these two parameters is useless in the proof, so we avoid using θ, σ in our scheme. This modification does not decrease the efficiency, and results in a more elegant structure.

Here we assume messages are from the target group G_T^2 and identities are from $\mathbb{Z}_p$.

$Setup(1^\lambda)$: The setup algorithm runs $\mathcal{G}(1^\lambda)$ to obtain (p, G, G_T, e). It samples random dual orthonormal basis $(\mathbb{D}, \mathbb{D}^*) \xleftarrow{R} Dual(\mathbb{Z}_p^6)$, chooses random $\alpha_1, \alpha_2 \in \mathbb{Z}_p$ and computes $\Omega_1 = e(g,g)^{\alpha_1 \mathbf{d}_1 \cdot \mathbf{d}_1^*}$ and $\Omega_2 = e(g,g)^{\alpha_2 \mathbf{d}_1 \cdot \mathbf{d}_1^*}$. The public parameters are: $PP = (G, p, \Omega_1, \Omega_2, g^{\mathbf{d}_1}, g^{\mathbf{d}_2}, g^{\mathbf{d}_3}, g^{\mathbf{d}_4})$. The master secret key is $Msk = (g^{\mathbf{d}_1^*}, g^{\alpha_1 \mathbf{d}_1^*}, g^{\mathbf{d}_2^*}, g^{\mathbf{d}_3^*}, g^{\alpha_2 \mathbf{d}_3^*}, g^{\mathbf{d}_4^*})$.

$Keygen(Msk, ID)$: The key generation algorithm chooses random $r_1, r_2, r_3, r_4 \in \mathbb{Z}_p$ and sets the secret key Sk_{ID} as:

$$K_1 = g^{(\alpha_1 + r_1 ID)\mathbf{d}_1^* - r_1 \mathbf{d}_2^* + r_2 ID \mathbf{d}_3^* - r_2 \mathbf{d}_4^*},$$

$$K_2 = g^{r_3 ID \mathbf{d}_1^* - r_3 \mathbf{d}_2^* + (\alpha_2 + r_4 ID)\mathbf{d}_3^* - r_4 \mathbf{d}_4^*}.$$

$Encrypt(PP, M_1 \| M_2, ID)$: The encryption algorithm chooses random $s_1, s_2 \in \mathbb{Z}_p$ and computes the ciphertext C as:

$$C_1 = M_1 \Omega_1^{s_1}, C_2 = M_2 \Omega_2^{s_2}, C_3 = g^{s_1 \mathbf{d}_1 + s_1 ID \mathbf{d}_2 + s_2 \mathbf{d}_3 + s_2 ID \mathbf{d}_4}.$$

$Decrypt(C, Sk_{ID})$: The decryption algorithm computes the message as:

$$M_1 = C_1 / e_n(C_3, K_1), M_2 = C_2 / e_n(C_3, K_2).$$

Correctness can be easily verified since $e_n(C_3, K_1) = \Omega_1^{s_1}, e(C_3, K_2) = \Omega_2^{s_2}$. Here $\mathbf{d}_5$ and $\mathbf{d}_6$ are used in the proof to form semi-functional ciphertexts and keys.

4.2 Security Proof

In the security proof of our IBE scheme, we use *semi-functional ciphertexts* and *semi-functional keys*, which are widely used in previous literatures [12, 17, 10, 15].

Semi-functional Ciphertexts. To create a semi-functional ciphertext, we first choose random $s_1, s_2, , z_1, z_2 \in \mathbb{Z}_p$ and set

$$C_1 = M_1\Omega_1^{s_1}, C_2 = M_2\Omega_2^{s_2}, C_3 = g^{s_1\mathbf{d}_1+s_1ID\mathbf{d}_2+s_2\mathbf{d}_3+s_2ID\mathbf{d}_4+z_1\mathbf{d}_5+z_2\mathbf{d}_6}.$$

Semi-functional Keys. To create a semi-functional key, we first choose random $r_1, r_2, r_3, r_4, t_1, t_2, t_3, t_4 \in \mathbb{Z}_p$ and set:

$$K_1 = g^{(\alpha_1+r_1ID)\mathbf{d}_1^*-r_1\mathbf{d}_2^*+r_2ID\mathbf{d}_3^*-r_2\mathbf{d}_4^*+t_1\mathbf{d}_5^*+t_2\mathbf{d}_6^*},$$

$$K_2 = g^{r_3ID\mathbf{d}_1^*-r_3\mathbf{d}_2^*+(\alpha_2+r_4ID)\mathbf{d}_3^*-r_4\mathbf{d}_4^*+t_3\mathbf{d}_5^*+t_4\mathbf{d}_6^*}.$$

We can see that a normal ciphertext can be correctly decrypted by a semi-functional key, and a semi-functional ciphertext can be correctly decrypted by a normal key, but when a semi-functional key is used to decrypt a semi-functional ciphertext, we will get the blinding factor multiplied by the additional term $e(g,g)^{\mathbf{d}_1\cdot\mathbf{d}_1^*(t_1z_1+t_2z_2)}$ and $e(g,g)^{\mathbf{d}_1\cdot\mathbf{d}_1^*(t_3z_1+t_4z_2)}$.

Theorem 1. *If DLIN assumption holds, then our IBE scheme is fully secure.*

Proof. To prove the security of our scheme, we define a sequence of games that any PPT adversary can not tell the difference between two adjacent games. Let q denote the number of key extraction queries that the adversary makes during the whole game.

$Game_{Real}$**:** the real fully security game.
$Game_0$**:** the same as $Game_{Real}$ except that the challenge ciphertext is semi-functional.
$Game_j$**:** for j from 1 to q, $Game_j$ is like $Game_0$ except that the first j key extraction queries are answered with semi-functional keys. The rest of the keys are normally generated.
$Game_{Final}$**:** the same as $Game_q$, except that the challenge ciphertext is a semi-functional encryption of a random message.

Let $Adv_{\mathcal{A}}^{Real}$ denote $\mathcal{A}$'s advantage in $Game_{Real}$, $Adv_{\mathcal{A}}^{i}$ denote $\mathcal{A}$'s advantage in $Game_i$ and $Adv_{\mathcal{A}}^{Final}$ denote $\mathcal{A}$'s advantage in $Game_{Final}$. It is clear that $Adv_{\mathcal{A}}^{Final} = 0$.

Lemma 2. *Suppose that there exists a PPT adversary $\mathcal{A}$ such that $Adv_{\mathcal{A}}^{Real} - Adv_{\mathcal{A}}^{0} = \epsilon$, then there exists a PPT adversary $\mathcal{B}$ with advantage ϵ in breaking the subspace assumption, with $k = 2$ and $n = 6$.*

Proof. $\mathcal{B}$ receives

$$D = (g^{\mathbf{b}_1}, ..., g^{\mathbf{b}_4}, g^{\eta\mathbf{b}_1^*}, g^{\eta\mathbf{b}_2^*}, g^{\beta\mathbf{b}_3^*}, g^{\beta\mathbf{b}_4^*}, g^{\mathbf{b}_5^*}, g^{\mathbf{b}_6^*}, U_1, U_2, \mu_3),$$

along with T_1, T_2, and its task is to decide whether T_1, T_2 is independent of $g^{\mathbf{b}_5^*}, g^{\mathbf{b}_6^*}$. $\mathcal{B}$ picks random invertible matrix $A \in \mathbb{Z}_p^{2\times 2}$. We define dual orthonormal bases $\mathbb{F}, \mathbb{F}^*$ as follows:

$$\mathbf{f}_1 = \eta\mathbf{b}_1^*, \mathbf{f}_2 = \eta\mathbf{b}_2^*, \mathbf{f}_3 = \beta\mathbf{b}_3^*, \mathbf{f}_4 = \beta\mathbf{b}_4^*, \mathbf{f}_5 = \mathbf{b}_5^*, \mathbf{f}_6 = \mathbf{b}_6^*,$$

$$\mathbf{f}_1^* = \eta^{-1}\mathbf{b}_1, \mathbf{f}_2^* = \eta^{-1}\mathbf{b}_2, \mathbf{f}_3^* = \beta^{-1}\mathbf{b}_3, \mathbf{f}_4^* = \beta^{-1}\mathbf{b}_4, \mathbf{f}_5^* = \mathbf{b}_5, \mathbf{f}_6^* = \mathbf{b}_6.$$

Then $\mathcal{B}$ implicitly sets $\mathbb{D} = \mathbb{F}_A, \mathbb{D}^* = \mathbb{F}_A^*$, that is, for $i = 1, 2, 3, 4, \mathbf{d}_i = \mathbf{f}_i, \mathbf{d}_i^* = \mathbf{f}_i^*, (\mathbf{d}_5, \mathbf{d}_6) = (\mathbf{f}_5, \mathbf{f}_6)A, (\mathbf{d}_5^*, \mathbf{d}_6^*) = (\mathbf{f}_5^*, \mathbf{f}_6^*)(A^{-1})^T$. Following from Lemma 1 in [10], we get that $(\mathbb{D}, \mathbb{D}^*)$ is randomly distributed and reveals no information about A.

Next $\mathcal{B}$ chooses random α_1', α_2' and implicitly sets $\alpha_1 = \eta\alpha_1', \alpha_2 = \beta\alpha_2'$, then $\mathcal{B}$ can compute $e(g,g)^{\alpha_1\mathbf{d}_1\cdot\mathbf{d}_1^*} = e_n(g^{\mathbf{b}_1}, g^{\eta\mathbf{b}_1^*})^{\alpha_1'}, e(g,g)^{\alpha_2\mathbf{d}_1\cdot\mathbf{d}_1^*} = e_n(g^{\mathbf{b}_3}, g^{\beta\mathbf{b}_3^*})^{\alpha_2'}$. Thus $\mathcal{B}$ sets up the public parameters and sends them to $\mathcal{A}$. $\mathcal{B}$ only knows the $g^{\alpha_1\mathbf{d}_1^*}, g^{\alpha_2\mathbf{d}_3^*}$ part of the master secret key and $g^{\eta\mathbf{d}_1^*}, g^{\eta\mathbf{d}_2^*}, g^{\beta\mathbf{d}_3^*}, g^{\beta\mathbf{d}_4^*}$. When $\mathcal{A}$ submits a key extraction query ID, $\mathcal{B}$ first chooses r_1', r_2', r_3', r_4' and implicitly sets $r_1 = \eta r_1', r_2 = \beta r_2', r_3 = \eta r_3', r_4 = \beta r_4'$. $\mathcal{B}$ sets the secret key as:

$$K_1 = g^{(\alpha_1' + r_1' ID)\eta\mathbf{d}_1^* - r_1'\eta\mathbf{d}_2^* + r_2' ID\beta\mathbf{d}_3^* - r_2'\beta\mathbf{d}_4^*},$$

$$K_2 = g^{r_3' ID\eta\mathbf{d}_1^* - r_3'\eta\mathbf{d}_2^* + (\alpha_2' + r_4' ID)\beta\mathbf{d}_3^* - r_4'\beta\mathbf{d}_4^*}.$$

At some point, $\mathcal{A}$ sends $\mathcal{B}$ the challenge identity ID^* and $(\mathbf{M_0}, \mathbf{M_1})$. $\mathcal{B}$ chooses a random bit $b \in \{0, 1\}$ and computes the challenge ciphertext as follows:

$$C_1 = M_{b,1} e_n(T_1, g^{\mathbf{b_1}})^{\alpha_1'}, C_2 = M_{b,2} e_n(T_1, g^{\mathbf{b_3}})^{\alpha_2'}, C_3 = T_1 (T_2)^{ID^*}.$$

and gives the answer to $\mathcal{A}$.

If $(T_1, T_2) = (V_1, V_2)$, then the respond is a normal ciphertext with $s_1 = \tau_1, s_2 = \tau_2$. If $(T_1, T_2) = (W_1, W_2)$, then the respond is a semi-functional ciphertext with $s_1 = \tau_1, s_2 = \tau_2$, and $(z_1, z_2)^T = \tau_3 A^{-1}(1, ID^*)^T$. Thus when $(T_1, T_2) = (V_1, V_2)$ we properly simulate $Game_{Real}$ and when $(T_1, T_2) = (W_1, W_2)$ we simulate $Game_0$. So $\mathcal{B}$ can leverage $\mathcal{A}$'s advantage in distinguishing $Game_{Real}$ and $Game_0$ to achieve the same advantage against the subspace assumption. □

Lemma 3. *Suppose that there exists a PPT algorithm $\mathcal{A}$ such that $Adv_{\mathcal{A}}^{j-1} - Adv_{\mathcal{A}}^{j} = \epsilon$, then there exists a PPT algorithm $\mathcal{B}$ with advantage ϵ in breaking the subspace assumption with $k = 2$ and $n = 6$.*

To prove the lemma, we define $Game'_j$ as an intermediate game and let $Adv_{\mathcal{A}}^{j'}$ be $\mathcal{A}$'s advantage in $Game'_j$.

$Game'_j$: for j from 1 to q, $Game'_j$ is like $Game_{j-1}$ except that the j-th key query is answered by a temporary semi-functional key. A temporary semi-functional key is generated as follows: we first choose random $r_1, r_2, r_3, r_4, t_1, t_2 \in \mathbb{Z}_p$ and set:

$$K_1 = g^{(\alpha_1 + r_1 ID)\mathbf{d}_1^* - r_1\mathbf{d}_2^* + r_2 ID\mathbf{d}_3^* - r_2\mathbf{d}_4^* + t_1\mathbf{d}_5^* + t_2\mathbf{d}_6^*},$$

$$K_2 = g^{r_3 ID\mathbf{d}_1^* - r_3\mathbf{d}_2^* + (\alpha_2 + r_4 ID)\mathbf{d}_3^* - r_4\mathbf{d}_4^*}.$$

Note that half part of the temporary semi-functional key is generated like semi-functional keys, and the other half is like normal keys. Here we change the extractable key to semi-functional in 2 steps to make r_1, r_2, r_3, r_4 randomly distributed.

Lemma 4. *Suppose that there exists a PPT algorithm $\mathcal{A}$ such that $Adv_{\mathcal{A}}^{j-1} - Adv_{\mathcal{A}}^{j'} = \epsilon$, then there exists a PPT algorithm $\mathcal{B}$ with advantage ϵ in breaking the subspace assumption with $k = 2$ and $n = 6$.*

Proof. $\mathcal{B}$ receives

$$D = (g^{\mathbf{b}_1}, ..., g^{\mathbf{b}_4}, g^{\eta\mathbf{b}_1^*}, g^{\eta\mathbf{b}_2^*}, g^{\beta\mathbf{b}_3^*}, g^{\beta\mathbf{b}_4^*}, g^{\mathbf{b}_5^*}, g^{\mathbf{b}_6^*}, U_1, U_2, \mu_3),$$

along with T_1, T_2, and its task is to decide whether T_1, T_2 is independent of $g^{\mathbf{b}_5^*}, g^{\mathbf{b}_6^*}$. $\mathcal{B}$ picks random invertible matrix $A \in \mathbb{Z}_p^{2\times 2}$. We implicitly set dual orthonormal bases $\mathbb{D} = \mathbb{B}_A, \mathbb{D}^* = \mathbb{B}_A^*$, that is, for $i = 1, 2, 3, 4, \mathbf{d}_i = \mathbf{b}_i, \mathbf{d}_i^* = \mathbf{b}_i^*, (\mathbf{d}_5, \mathbf{d}_6) = (\mathbf{b}_5, \mathbf{b}_6)A, (\mathbf{d}_5^*, \mathbf{d}_6^*) = (\mathbf{b}_5^*, \mathbf{b}_6^*)(A^{-1})^T$. Following from Lemma 1 introduced in [10], we get that $(\mathbb{D}, \mathbb{D}^*)$ is randomly distributed and reveals no information about A.

Next $\mathcal{B}$ chooses random α_1', α_2' and implicitly sets $\alpha_1 = \eta\alpha_1', \alpha_2 = \beta\alpha_2'$, then $\mathcal{B}$ can compute $e(g,g)^{\alpha_1 \mathbf{d}_1 \cdot \mathbf{d}_1^*} = e_n(g^{\mathbf{b}_1}, g^{\eta \mathbf{b}_1^*})^{\alpha_1'}, e(g,g)^{\alpha_2 \mathbf{d}_1 \cdot \mathbf{d}_1^*} = e_n(g^{\mathbf{b}_3}, g^{\beta \mathbf{b}_3^*})^{\alpha_2'}$, so $\mathcal{B}$ sets up the public parameters and sends them to $\mathcal{A}$.

When $\mathcal{A}$ submits key extraction queries:

For $i < j$, $\mathcal{B}$ can answer it since $\mathcal{B}$ can create normal keys as in Lemma 2 and knows $\mathbf{d}_5^*, \mathbf{d}_6^*$.
For $i > j$, $\mathcal{B}$ can answer it as in Lemma 2.
For $i = j$, $\mathcal{B}$ first chooses random $r_3', r_4' \in \mathbb{Z}_p$ and sets:

$$K_1 = g^{\alpha_1' \eta \mathbf{d}_1^*} (T_1^{ID_j}) T_2^{-1}, K_2 = g^{r_3' ID_j \eta \mathbf{d}_1^* - r_3' \eta \mathbf{d}_2^* + (\alpha_2' + r_4' ID_j)\beta \mathbf{d}_3^* - r_4' \beta \mathbf{d}_4^*}.$$

At some point, $\mathcal{A}$ sends $\mathcal{B}$ the challenge identity ID^* and $(\mathbf{M_0}, \mathbf{M_1})$. $\mathcal{B}$ chooses a random bit $b \in \{0,1\}$ and computes the challenge ciphertext as follows:

$$C_1 = M_{b,1} e_n(U_1, g^{\eta \mathbf{b_1}^*})^{\alpha_1'}, C_2 = M_{b,2} e_n(U_1, g^{\beta \mathbf{b_3}^*})^{\alpha_2'}, C_3 = U_1 (U_2)^{ID^*}.$$

and gives the answer to $\mathcal{A}$. Here we implicitly set $s_1 = \mu_1, s_2 = \mu_2, (z_1, z_2)^T = \mu_3 A^{-1}(1, ID^*)^T$.

If $(T_1, T_2) = (V_1, V_2)$, then the respond is a normal key with $r_1 = \tau_1\eta, r_2 = \tau_2\beta$. If $(T_1, T_2) = (W_1, W_2)$, then the respond is a semi-functional key with $r_1 = \tau_1\eta, r_2 = \tau_2\beta$, and $(t_1, t_2)^T = \tau_3 A^{-1}(1, ID_j)^T$. Since $ID_j \neq ID^*$, and A is information-theoretically hidden from $(\mathbb{D}, \mathbb{D}^*)$, challenge ciphertext and the j-th key query are randomly distributed in $\mathcal{A}'s$ view.

When $(T_1, T_2) = (V_1, V_2)$ we properly simulate $Game_{j-1}$ and when $(T_1, T_2) = (W_1, W_2)$ we simulate $Game_{j'}$. So $\mathcal{B}$ can leverage $\mathcal{A}$'s advantage in distinguishing $Game_{j-1}$ and $Game_{j'}$ to achieve the same advantage against the subspace assumption. □

Lemma 5. *Suppose that there exists a PPT algorithm $\mathcal{A}$ such that $Adv_{\mathcal{A}}^{j'} - Adv_{\mathcal{A}}^{j} = \epsilon$, then there exists a PPT algorithm $\mathcal{B}$ with advantage ϵ in breaking the subspace assumption with $k = 2$ and $n = 6$.*

Proof. The proof of this lemma is similar to that of Lemma 4 and we put the concret proof in our appendix .

Lemma 6. *Suppose that there exists a PPT algorithm $\mathcal{A}$ such that $Adv^{q}_{\mathcal{A}} - Adv^{Final}_{\mathcal{A}} = \epsilon$. Then there exists a PPT algorithm $\mathcal{B}$ with advantage ϵ in breaking the subspace assumption with $n = 6$ and $k = 1$.*

In order to prove this lemma, we define $Game_{q_a}, Game_{q_b}, Game_{q_c}$ as intermediate games:

$Game_{q_a}$: It is exactly like $Game_q$ except that in the C^*_3 part of the challenge ciphertext, the coefficient of $\mathbf{d}_2$ is changed to a random value in $\mathbb{Z}_p$ instead of $s_1 ID^*$.

$Game_{q_b}$: It is exactly like $Game_{q_a}$ except that in the C^*_3 part of the challenge ciphertext is independent of s_1.

$Game_{q_c}$: It is exactly like $Game_{q_b}$ except that in the C^*_3 part of the challenge ciphertext, the coefficient of $\mathbf{d}_4$ is changed to a random value in $\mathbb{Z}_p$ instead of $s_2 ID^*$.

We denote the advantage in these games as $Adv^{q_a}_{\mathcal{A}}, Adv^{q_b}_{\mathcal{A}}, Adv^{q_c}_{\mathcal{A}}$.

Lemma 7. *Suppose that there exists a PPT algorithm $\mathcal{A}$ such that $Adv^{q}_{\mathcal{A}} - Adv^{q_a}_{\mathcal{A}} = \epsilon$, then there exists a PPT algorithm $\mathcal{B}$ with advantage ϵ in breaking the subspace assumption with $k = 1$ and $n = 6$.*

Proof. $\mathcal{B}$ receives

$$D = (g^{\mathbf{b}_1}, g^{\mathbf{b}_2}, g^{\mathbf{b}_4}, ..., g^{\mathbf{b}_6}, g^{\eta \mathbf{b}^*_1}, g^{\beta \mathbf{b}^*_2}, g^{\mathbf{b}^*_3}, ..., g^{\mathbf{b}^*_6}, U_1, \mu_3),$$

along with T_1, and its task is to decide whether T_1 is independent of $g^{\mathbf{b}^*_3}$. $\mathcal{B}$ implicitly sets dual orthonormal bases as follows:

$$\mathbf{d}_1 = \mathbf{b}^*_6, \mathbf{d}_2 = \mathbf{b}^*_3, \mathbf{d}_3 = \mathbf{b}^*_5, \mathbf{d}_4 = \mathbf{b}^*_4, \mathbf{d}_5 = \mathbf{b}^*_2, \mathbf{d}_6 = \mathbf{b}^*_1,$$

$$\mathbf{d}^*_1 = \mathbf{b}_6, \mathbf{d}^*_2 = \mathbf{b}_3, \mathbf{d}^*_3 = \mathbf{b}_5, \mathbf{d}^*_4 = \mathbf{b}_4, \mathbf{d}^*_5 = \mathbf{b}_2, \mathbf{d}^*_6 = \mathbf{b}_1.$$

Next $\mathcal{B}$ chooses random $\alpha_1, \alpha_2 \in \mathbb{Z}_p$ and the public parameters can be correctly computed. Also $\mathcal{B}$ can compute the master secret key except $g^{\mathbf{d}^*_2}$.

When $\mathcal{A}$ submits key extraction queries, $\mathcal{B}$ first chooses random r'_1, r_2, r'_3, $r_4, t'_1, t'_2, t'_3, t'_4 \in \mathbb{Z}_p$ and sets:

$$K_1 = (U_1)^{-r'_1} g^{(\alpha_1+\mu_3 r'_1 ID)\mathbf{d}^*_1 + r_2 ID\mathbf{d}^*_3 - r_2\mathbf{d}^*_4 + t'_1\mathbf{d}^*_5 + t'_2\mathbf{d}^*_6},$$

$$K_2 = (U_1)^{-r'_3} g^{\mu_3 r'_3 ID\mathbf{d}^*_1 + (\alpha_2+r_4 ID)\mathbf{d}^*_3 - r_4\mathbf{d}^*_4 + t'_3\mathbf{d}^*_5 + t'_4\mathbf{d}^*_6}.$$

Here we implicitly set $r_1 = \mu_3 r'_1, r_3 = \mu_3 r'_3, t_1 = t'_1 - \mu_2 r'_1, t_2 = t'_2 - \mu_1 r'_1, t_3 = t'_3 - \mu_2 r'_3, t_4 = t'_4 - \mu_1 r'_3$.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ the challenge identity ID^* and $(\mathbf{M_0}, \mathbf{M_1}), \mathcal{B}$ chooses a random bit $b \in \{0, 1\}, s_1, s_2 \in \mathbb{Z}_p$ and computes the challenge ciphertext as follows:

$$C_1 = \Omega_1^{s_1} M_{b,1}, C_2 = \Omega_2^{s_2} M_{b,2}, C_3 = g^{s_1\mathbf{d}_1 + s_1 ID^*\mathbf{d}_2 + s_2\mathbf{d}_3 + s_2 ID^*\mathbf{d}_4} T_1.$$

Then $\mathcal{B}$ gives the answer to $\mathcal{A}$. Here we implicitly set $z_1 = \tau_2\beta, z_2 = \tau_1\eta$.

If $T_1 = V_1$ we properly simulate $Game_q$ and when $T_1 = W_1$ we simulate $Game_{q_a}$. So $\mathcal{B}$ can leverage $\mathcal{A}$'s advantage in distinguishing $Game_q$ and $Game_{q_a}$ to achieve the same advantage against the subspace assumption. □

Lemma 8. *Suppose that there exists a PPT algorithm $\mathcal{A}$ such that $Adv^{q_a}_{\mathcal{A}} - Adv^{q_b}_{\mathcal{A}} = \epsilon$, then there exists a PPT algorithm $\mathcal{B}$ with advantage ϵ in breaking the subspace assumption with $k = 1$ and $n = 6$.*

Proof. $\mathcal{B}$ receives

$$D = (g^{\mathbf{b}_1}, g^{\mathbf{b}_2}, g^{\mathbf{b}_4}, ..., g^{\mathbf{b}_6}, g^{\eta\mathbf{b}^*_1}, g^{\beta\mathbf{b}^*_2}, g^{\mathbf{b}^*_3}, ..., g^{\mathbf{b}^*_6}, U_1, \mu_3),$$

along with T_1, and its task is to decide whether T_1 is independent of $g^{\mathbf{b}^*_3}$. $\mathcal{B}$ implicitly sets dual orthonormal bases as follows:

$$\mathbf{d}_1 = \mathbf{b}^*_3, \mathbf{d}_2 = \mathbf{b}^*_4, \mathbf{d}_3 = \mathbf{b}^*_5, \mathbf{d}_4 = \mathbf{b}^*_6, \mathbf{d}_5 = \mathbf{b}^*_1, \mathbf{d}_6 = \mathbf{b}^*_2,$$

$$\mathbf{d}^*_1 = \mathbf{b}_3, \mathbf{d}^*_2 = \mathbf{b}_4, \mathbf{d}^*_3 = \mathbf{b}_5, \mathbf{d}^*_4 = \mathbf{b}_6, \mathbf{d}^*_5 = \mathbf{b}_1, \mathbf{d}^*_6 = \mathbf{b}_2.$$

Next $\mathcal{B}$ chooses random $\alpha'_1, \alpha_2 \in \mathbb{Z}_p$ and implicitly sets $\alpha_1 = \alpha'_1\mu_3$, so the public parameters can be correctly computed $(e(g, g)^{\alpha_1\mathbf{d}_1\cdot\mathbf{d}^*_1} = e_n(g^{\mathbf{b}_4}, g^{\mathbf{b}^*_4})^{\alpha_1})$. Also $\mathcal{B}$ can compute the master secret key except $g^{\mathbf{d}^*_1}, g^{\alpha_1\mathbf{d}^*_1}$.

When $\mathcal{A}$ submits key extraction queries, $\mathcal{B}$ first chooses random $r'_1, r_2, r'_3, r_4, t'_1, t'_2, t'_3, t'_4 \in \mathbb{Z}_p$ and sets:

$$K_1 = (U_1)^{\alpha'_1 + r'_1 ID} g^{-r'_1 \mu_3 \mathbf{d}^*_2 + r_2 ID \mathbf{d}^*_3 - r_2 \mathbf{d}^*_4 + t'_1 \mathbf{d}^*_5 + t'_2 \mathbf{d}^*_6},$$

$$K_2 = (U_1)^{r'_3 ID} g^{-\mu_3 r'_3 \mathbf{d}^*_2 + (\alpha_2 + r_4 ID)\mathbf{d}^*_3 - r_4 \mathbf{d}^*_4 + t'_3 \mathbf{d}^*_5 + t'_4 \mathbf{d}^*_6}.$$

Here we implicitly set $r_1 = \mu_3 r'_1, r_3 = \mu_3 r'_3, t_1 = t'_1 + \mu_1(\alpha'_1 + r'_1 ID), t_2 = t'_2 + \mu_2(\alpha'_1 + r'_1 ID), t_3 = t'_3 + \mu_1 ID r'_3, t_4 = t'_4 + \mu_2 ID r'_3$.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ the challenge identity ID^* and $(\mathbf{M_0}, \mathbf{M_1})$, $\mathcal{B}$ chooses a random bit $b \in \{0, 1\}, s_1, s_2, w \in \mathbb{Z}_p$ and computes the challenge ciphertext as follows:

$$C_1 = \Omega_1^{s_1} M_{b,1}, C_2 = \Omega_2^{s_2} M_{b,2}, C_3 = g^{s_1 \mathbf{d}_1 + w \mathbf{d}_2 + s_2 \mathbf{d}_3 + s_2 ID^* \mathbf{d}_4} T_1.$$

and gives the answer to $\mathcal{A}$. Here we implicitly set $z_1 = \tau_1 \eta, z_2 = \tau_2 \beta$.

If $T_1 = V_1$ we properly simulate $Game_{q_a}$ and when $T_1 = W_1$ we simulate $Game_{q_b}$. So $\mathcal{B}$ can leverage $\mathcal{A}$'s advantage in distinguishing $Game_{q_a}$ and $Game_{q_b}$ to achieve the same advantage against the subspace assumption. □

Lemma 9. *Suppose that there exists a PPT algorithm $\mathcal{A}$ such that $Adv^{q_b}_{\mathcal{A}} - Adv^{q_c}_{\mathcal{A}} = \epsilon$, then we can construct a PPT algorithm $\mathcal{B}$ with advantage ϵ in breaking the subspace assumption with $k = 1$ and $n = 6$.*

Lemma 10. *Suppose that there exists a PPT algorithm $\mathcal{A}$ such that $Adv^{q_c}_{\mathcal{A}} - Adv^{Final}_{\mathcal{A}} = \epsilon$, then there exists a PPT algorithm $\mathcal{B}$ with advantage ϵ in breaking the subspace assumption with $k = 1$ and $n = 6$.*

The proof of Lemma 9 and Lemma 10 is similar to that of Lemma 7 and Lemma 8, so we omit it here and put it in the appendix.

The previous lemmata show that the real security game is indistinguishable from $Game_{Final}$, in which the value of b is information-theoretically hidden from the attacker, hence the attacker can only get negligible advantage in breaking the security of our IBE scheme. □

4.3 Anonymity

We note that in the final game, the challenge ciphertext is independent of the challenge identity, so our scheme is anonymous. Lewko's IBE scheme is

anonymous for the same reason. Anonymous IBE [5] is a useful component to construct public key encryption with keyword search (PEKS) schemes [1].

5 Conclusion

In this paper, we improve the IBE scheme presented by Lewko in [10], and get a fully secure anonymous IBE scheme in the prime order setting that has a better message-ciphertext rate. Similar to Lewko's scheme, we use dual pairing vector space in prime order bilinear groups to realize the canceling and parameter hiding property. The security of our scheme is based on the subspace assumption, which can be reduced to the decisional linear assumption. We use the dual system encryption in the security proof.

References

1. M. Abdalla, M. Bellare, D. Catalano, and E Kiltz. Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions. In *CRYPTO 2005, LNCS 3621*, pages 205–222, Berlin, Heidelberg, 2005. Springer-Verlag.
2. D. Boneh and X. Boyen. Secure identity based encryption without random oracles. In *CRYPTO 2004, LNCS 3152*, pages 443–459, Berlin, Heidelberg, 2004. Springer-Verlag.
3. D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO 2001, LNCS 2139*, pages 213–229, Berlin, Heidelberg, 2001. Springer-Verlag.
4. D. Boneh, A. Sahai, and B. Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In *EUROCRYPT 2006, LNCS 4004*, pages 573–592, Berlin, Heidelberg, 2006. Springer-Verlag.
5. X. Boyen and B. Waters. Anonymous hierarchical identity-based encryption (without random oracles). In *CRYPTO 2006, LNCS 4117*, pages 290–307, Berlin, Heidelberg, 2006. Springer-Verlag.
6. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT 2004, LNCS 3027*, pages 207–222, Berlin, Heidelberg, 2004. Springer-Verlag.
7. C. Cocks. An identity based encryption scheme based on quadratic residues. In *Cryptography and Coding 2001, LNCS 2260*, pages 360–363, Berlin Heidelberg, 2001. Springer-Verlag.
8. D.M Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In *EUROCRYPT 2010, LNCS 6110*, pages 44–61, Berlin, Heidelberg, 2010. Springer-Verlag.

9. J. Groth, R. Ostrovsky, and A. Sahai. Non-interactive zaps and new techniques for nizk. In *CRYPTO 2006, LNCS 4117*, pages 97–111, Berlin, Heidelberg, 2006. Springer-Verlag.
10. A. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In *EUROCRYPT 2012, LNCS 7237*, pages 318–335, Berlin, Heidelberg, 2012. Springer-Verlag.
11. A. Lewko, T. Okamoto, and A. Sahai. Fully secure functional encryption :attribute based encryption and (hierarichical) inner product encryption. In *EUROCRYPT 2010, LNCS 6110*, pages 62–91, Berlin, Heidelberg, 2010. Springer-Verlag.
12. A. Lewko and B. Waters. New techniques for dual system encryption and fully secure hibe with short ciphertexts. In *Theory of Cryptography, LNCS 5978*, pages 455–479, Berlin, Heidelberg, 2010. Springer-Verlag.
13. T. Okamoto and K. Takashima. Homomorphic encryption and signatures from vector decomposition. In *Paring 2008, LNCS 5209*, pages 57–74, Berlin, Heidelberg, 2008. Springer-Verlag.
14. T. Okamoto and K. Takashima. Hierarchical predicate encryption for inner products. In *ASIACRYPT 2009, LNCS 5912*, pages 214–231, Berlin, Heidelberg, 2009. Springer-Verlag.
15. C. Ramanna1, S. Chatterjee, and R. Sarkar. Variants of waters dual system primitives using asymmetric pairings. In *PKC 2008, LNCS 7293*, pages 298–315, Berlin, Heidelberg, 2012. Springer-Verlag.
16. A. Shamir. Identity-based cryptosystems and signature schemes. In G.R.Blakley and D.Chaum, editors, *Crypto 1984, LNCS 196*, pages 47–53. Springer, 1984. Proceedings of CRYPTO.
17. B. Waters. Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In *CRYPTO 2009, LNCS 5677*, pages 519–636, Berlin, Heidelberg, 2009. Springer-Verlag.

Appendix

Proof of Lemma 5

Proof. $\mathcal{B}$ receives

$$D = (g^{\mathbf{b}_1}, ..., g^{\mathbf{b}_4}, g^{\eta\mathbf{b}_1^*}, g^{\eta\mathbf{b}_2^*}, g^{\beta\mathbf{b}_3^*}, g^{\beta\mathbf{b}_4^*}, g^{\mathbf{b}_5^*}, g^{\mathbf{b}_6^*}, U_1, U_2, \mu_3),$$

along with T_1, T_2, and its task is to decide whether T_1, T_2 is independent of $g^{\mathbf{b}_5^*}, g^{\mathbf{b}_6^*}$. $\mathcal{B}$ picks random invertible matrix $A \in \mathbb{Z}_p^{2\times 2}$. We implicitly set dual orthonormal bases $\mathbb{D} = \mathbb{B}_A, \mathbb{D}^* = \mathbb{B}_A^*$, that is, for $i = 1, 2, 3, 4, \mathbf{d}_i = \mathbf{b}_i, \mathbf{d}_i^* = \mathbf{b}_i^*$,

$$(\mathbf{d}_5, \mathbf{d}_6) = (\mathbf{b}_5, \mathbf{b}_6)A, (\mathbf{d}_5^*, \mathbf{d}_6^*) = (\mathbf{b}_5^*, \mathbf{b}_6^*)(A^{-1})^T.$$

Following from the lemma introduce in [10], we get that $(\mathbb{D}, \mathbb{D}^*)$ is random distributed and reveals no information about A.

Next $\mathcal{B}$ chooses random α_1', α_2' and implicitly sets $\alpha_1 = \eta\alpha_1', \alpha_2 = \beta\alpha_2'$, then $\mathcal{B}$ can compute $e(g,g)^{\alpha_1 \mathbf{d}_1 \cdot \mathbf{d}_1^*} = e_n(g^{\mathbf{b}_1}, g^{\eta \mathbf{b}_1^*})^{\alpha_1'}, e(g,g)^{\alpha_2 \mathbf{d}_1 \cdot \mathbf{d}_1^*} = e_n(g^{\mathbf{b}_3}, g^{\beta \mathbf{b}_3^*})^{\alpha_2'}$, so $\mathcal{B}$ knows the public parameters and can give it to $\mathcal{A}$.

When $\mathcal{A}$ submits key extraction queries:

For $i < j$, $\mathcal{B}$ can answer it since $\mathcal{B}$ can create normal keys as in Lemma 2 and knows $\mathbf{d}_5^*, \mathbf{d}_6^*$.
For $i > j$, $\mathcal{B}$ can answer it as in Lemma 2.
For $i = j$, $\mathcal{B}$ first chooses random $r_1', r_2', t_1, t_2 \in \mathbb{Z}_p$ and sets:

$$K_1 = g^{(\alpha_1' + r_1' ID_j)\eta \mathbf{d}_1^* - r_1' \eta \mathbf{d}_2^* + r_2' ID_j \beta \mathbf{d}_3^* - r_2' \beta \mathbf{d}_4^* + t_1 \mathbf{d}_5^* + t_2 \mathbf{d}_6^*},$$

$$K_2 = g^{\alpha_2' \beta \mathbf{d}_3^*} (T_1)^{ID_j} T_2^{-1}.$$

At some point, $\mathcal{A}$ sends $\mathcal{B}$ the challenge identity ID^* and $(\mathbf{M_0}, \mathbf{M_1})$. $\mathcal{B}$ chooses a random bit $b \in \{0,1\}$ and computes the challenge ciphertext as follows:

$$C_1 = M_{b,1} e_n(U_1, g^{\eta \mathbf{b_1}^*})^{\alpha_1'}, C_2 = M_{b,2} e_n(U_1, g^{\beta \mathbf{b_3}^*})^{\alpha_2'}, C_3 = U_1 (U_2)^{ID^*}.$$

and gives the answer to $\mathcal{A}$. Here we implicitly set $s_1 = \mu_1, s_2 = \mu_2, (z_1, z_2)^T = \mu_3 A^{-1} (1, ID^*)^T$.

If $(T_1, T_2) = (V_1, V_2)$, then the respond is a normal key with $r_3 = \tau_1 \eta, r_4 = \tau_2 \beta$. If $(T_1, T_2) = (W_1, W_2)$, then the respond is a semi-functional key with $r_3 = \tau_1 \eta, r_4 = \tau_2 \beta$, and $(t_1, t_2)^T = \tau_3 A^{-1} (1, ID_j)^T$. Since $ID_j \neq ID^*$, and A is information-theoretically hidden from $(\mathbb{D}, \mathbb{D}^*)$, challenge ciphertext and the j-th key query are randomly distributed in $\mathcal{A}'s$ view.

Thus when $(T_1, T_2) = (V_1, V_2)$ we properly simulate $Game_{j-1}$ and when $(T_1, T_2) = (W_1, W_2)$ we simulate $Game_{j'}$. So $\mathcal{B}$ can leverage $\mathcal{A}$'s advantage in distinguishing $Game_{j'}$ and $Game_j$ to achieve the same advantage against the subspace assumption. □

Proof of Lemma 9

Proof. $\mathcal{B}$ receives

$$D = (g^{\mathbf{b}_1}, g^{\mathbf{b}_2}, g^{\mathbf{b}_4}, ..., g^{\mathbf{b}_6}, g^{\eta \mathbf{b}_1^*}, g^{\beta \mathbf{b}_2^*}, g^{\mathbf{b}_3^*}, ..., g^{\mathbf{b}_6^*}, U_1, \mu_3),$$

along with T_1, and its task is to decide whether T_1 is independent of $g^{\mathbf{b}_3^*}$. $\mathcal{B}$ implicitly sets dual orthonormal bases as follows:

$$\mathbf{d}_1 = \mathbf{b}_4^*, \mathbf{d}_2 = \mathbf{b}_5^*, \mathbf{d}_3 = \mathbf{b}_6^*, \mathbf{d}_4 = \mathbf{b}_3^*, \mathbf{d}_5 = \mathbf{b}_1^*, \mathbf{d}_6 = \mathbf{b}_2^*,$$

$$\mathbf{d}_1^* = \mathbf{b}_4, \mathbf{d}_2^* = \mathbf{b}_5, \mathbf{d}_3^* = \mathbf{b}_6, \mathbf{d}_4^* = \mathbf{b}_3, \mathbf{d}_5^* = \mathbf{b}_1, \mathbf{d}_6^* = \mathbf{b}_2.$$

Next $\mathcal{B}$ chooses random $\alpha_1, \alpha_2 \in \mathbb{Z}_p$ and the public parameters can be correctly computed. Also $\mathcal{B}$ can compute the master secret key except $g^{\mathbf{d}_4^*}$.

When $\mathcal{A}$ submits key extraction queries, $\mathcal{B}$ first chooses random $r_1, r_2', r_3, r_4', t_1', t_2', t_3', t_4' \in \mathbb{Z}_p$ and sets:

$$K_1 = (U_1)^{-r_2'} g^{(\alpha_1 + r_1 ID)\mathbf{d}_1^* - r_1 \mathbf{d}_2^* + \mu_3 r_2' ID \mathbf{d}_3^* + t_1' \mathbf{d}_5^* + t_2' \mathbf{d}_6^*},$$

$$K_2 = (U_1)^{-r_4'} g^{r_3 ID \mathbf{d}_1^* - r_3 \mathbf{d}_2^* + (\alpha_2 + \mu_3 r_4' ID)\mathbf{d}_3^* + t_3' \mathbf{d}_5^* + t_4' \mathbf{d}_6^*}.$$

Here we implicitly set $r_2 = \mu_3 r_2', r_4 = \mu_3 r_4', t_1 = t_1' - \mu_1 r_1', t_2 = t_2' - \mu_2 r_1', t_3 = t_3' - \mu_1 r_3', t_4 = t_4' - \mu_2 r_3'$.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ the challenge identity ID^* and $(\mathbf{M_0}, \mathbf{M_1})$, $\mathcal{B}$ chooses a random bit $b \in \{0,1\}, s_1, s_2, w_1, w_2 \in \mathbb{Z}_p$ and computes the challenge ciphertext as follows:

$$C_1 = \Omega_1^{s_1} M_{b,1}, C_2 = \Omega_2^{s_2} M_{b,2},$$

$$C_3 = g^{w_1 \mathbf{d}_1 + w_2 \mathbf{d}_2 + s_2 \mathbf{d}_3 + s_2 ID^* \mathbf{d}_4} T_1$$

and gives the answer to $\mathcal{A}$. Here we implicitly set $z_1 = \tau_1 \eta, z_2 = \tau_2 \beta$.

If $T_1 = V_1$ we properly simulate $Game_{q_b}$ and when $T_1 = W_1$ we simulate $Game_{q_c}$. So $\mathcal{B}$ can leverage $\mathcal{A}$'s advantage in distinguishing $Game_{q_b}$ and $Game_{q_c}$ to achieve the same advantage against the subspace assumption. □

Proof of Lemma 10

Proof. $\mathcal{B}$ receives

$$D = (g^{\mathbf{b}_1}, g^{\mathbf{b}_2}, g^{\mathbf{b}_4}, ..., g^{\mathbf{b}_6}, g^{\eta \mathbf{b}_1^*}, g^{\beta \mathbf{b}_2^*}, g^{\mathbf{b}_3^*}, ..., g^{\mathbf{b}_6^*}, U_1, \mu_3),$$

along with T_1, and its task is to decide whether T_1 is independent of $g^{\mathbf{b}_3^*}$. $\mathcal{B}$ implicitly sets dual orthonormal bases as follows:

$$\mathbf{d}_1 = \mathbf{b}_5^*, \mathbf{d}_2 = \mathbf{b}_6^*, \mathbf{d}_3 = \mathbf{b}_3^*, \mathbf{d}_4 = \mathbf{b}_4^*, \mathbf{d}_5 = \mathbf{b}_1^*, \mathbf{d}_6 = \mathbf{b}_2^*,$$

$$\mathbf{d}_1^* = \mathbf{b}_5, \mathbf{d}_2^* = \mathbf{b}_6, \mathbf{d}_3^* = \mathbf{b}_3, \mathbf{d}_4^* = \mathbf{b}_4, \mathbf{d}_5^* = \mathbf{b}_1, \mathbf{d}_6^* = \mathbf{b}_2.$$

Next $\mathcal{B}$ chooses random $\alpha_1, \alpha_2' \in \mathbb{Z}_p$ and implicitly sets $\alpha_2 = \alpha_2'\mu_3$, so the public parameters can be correctly computed. Also $\mathcal{B}$ can compute the master secret key except $g^{\mathbf{d}_3^*}, g^{\alpha_2 \mathbf{d}_3^*}$.

When $\mathcal{A}$ submits key extraction queries, $\mathcal{B}$ first chooses random r_1, r_2', r_3, $r_4', t_1', t_2', t_3', t_4' \in \mathbb{Z}_p$ and sets:

$$K_1 = (U_1)^{r_2' ID} g^{(\alpha_1 + r_1 ID)\mathbf{d}_1^* - r_1 \mathbf{d}_2^* - r_2' \mu_3 \mathbf{d}_4^* + t_1' \mathbf{d}_5^* + t_2' \mathbf{d}_6^*},$$

$$K_2 = (U_1)^{\alpha_2' + r_4' ID} g^{r_3 ID \mathbf{d}_1^* - r_3 \mathbf{d}_2^* - \mu_3 r_4' \mathbf{d}_4^* + t_3' \mathbf{d}_5^* + t_4' \mathbf{d}_6^*}.$$

Here we implicitly set $r_2 = \mu_3 r_2', r_4 = \mu_3 r_4', t_1 = t_1' + \mu_1 r_2' ID, t_2 = t_2' + \mu_2 r_2' ID, t_3 = t_3' + \mu_1(\alpha_2' + r_4' ID), t_4 = t_4' + \mu_2(\alpha_2' + r_4' ID)$.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ the challenge identity ID^* and $(\mathbf{M_0}, \mathbf{M_1})$, $\mathcal{B}$ chooses a random bit $b \in \{0, 1\}, s_1, s_2, w_1, w_2, w_3 \in \mathbb{Z}_p$ and computes the challenge ciphertext as follows:

$$C_1 = \Omega_1^{s_1} M_{b,1}, C_2 = \Omega_2^{s_2} M_{b,2}, C_3 = g^{w_1 \mathbf{d}_1 + w_2 \mathbf{d}_2 + s_2 \mathbf{d}_3 + w_3 \mathbf{d}_4} T_1.$$

and gives the answer to $\mathcal{A}$. Here we implicitly set $z_1 = \tau_1 \eta, z_2 = \tau_2 \beta$.

If $T_1 = V_1$ we properly simulate $Game_{q_c}$ and when $T_1 = W_1$ we simulate $Game_{q_{Final}}$. So $\mathcal{B}$ can leverage $\mathcal{A}$'s advantage in distinguishing $Game_{q_c}$ and $Game_{q_{Final}}$ to achieve the same advantage against the subspace assumption. □