Tightly Secure IBE under Constant-size Master Public Key

Jie Chen, Junqing Gong, Jian Weng

Tightly Secure IBE under Constant-size Master Public Key

Jie Chen[1⋆], Junqing Gong[2,3⋆⋆], and Jian Weng[4⋆⋆⋆]

[1] East China Normal University, Shanghai, China
[2] Shanghai Jiao Tong University, Shanghai, China
[3] Laboratoire LIP, École Normale Supérieure de Lyon, Lyon, France
[4] Jinan University, Guangzhou, China

Abstract. Chen and Wee [CRYPTO, 2013] proposed the first almost tightly and adaptively secure IBE in the standard model and left two open problems which called for a tightly secure IBE with (1) constant-size master public key and/or (2) constant security loss. In this paper, we propose an IBE scheme with *constant-size* master public key and *tighter* security reduction. This (partially) solves Chen and Wee's first open problem and makes progress on the second one. Technically, our IBE scheme is built based on Wee's petit IBE scheme [TCC, 2016] in the composite-order bilinear group whose order is product of four primes. The sizes of master public key, ciphertexts, and secret keys are not only constant but also nearly optimal as Wee's petit IBE. We can prove its adaptive security in the *multi-instance, multi-ciphertext* setting [PKC, 2015] based on the decisional subgroup assumption and a subgroup variant of DBDH assumption. The security loss is $O(\log q)$ where q is the upper bound of the *total* number of secret keys and challenge ciphertexts revealed to adversary in each *single* IBE instance. It's much smaller than those for all known adaptively secure IBE schemes in a *concrete* sense.

Keywords. identity based encryption; tight security; constant-size public key; composite-order group; Déjà Q technique

⋆ Email: S080001@e.ntu.edu.sg. Homepage: http://www.jchen.top. Supported by the National Natural Science Foundation of China (Nos. 61472142, 61632012) and the Science and Technology Commission of Shanghai Municipality (No. 14YF1404200). Part of this work was done while at École Normale Supérieure de Lyon in France.

⋆⋆ Email: junqing.gong@ens-lyon.fr. Supported by the French ANR ALAMBIC project (ANR-16-CE39-0006).

⋆⋆⋆ Email: cryptjweng@gmail.com. Supported by the National Natural Science Foundation of China (Nos. 61272413, 61472165, 61133014).

1 Introduction

In 1984, Shamir introduced the notion of *identity based encryptions* [Sha84] (IBE). The entire system is maintained by an authority called *Key Generation Center* (KGC) who publishes a master public key MPK and keeps the corresponding master secret key MSK. To encrypt a message to a user in the system, one only needs MPK and user's identity ID, which can be a descriptive tag such as email address. Each user receives his/her secret key SK for decryption from KGC which is produced using MSK according to the identity ID.

Boneh and Franklin, in their seminal work [BF01] in 2001, formulated the security notion of IBE and proposed a pairing-based IBE in the random oracle model [BR93]. Their security model has been accepted as standard model for IBE which ensures that a ciphertext for target identity ID^* reveals nothing of the plaintext even when adversary $\mathcal{A}$ holding MPK can obtain secret keys for any identity other than ID^*. We call it *adaptive security* in the paper. After that, a series of work were devoted to constructing IBE schemes in the standard model (i.e., without random oracle) including Boneh and Boyen's IBE [BB04a] in the selective model[1], Boneh and Boyen's IBE [BB04b] with huge security loss, Waters' IBE [Wat05] with large MPK, and Gentry's IBE [Gen06] based on a q-type assumption[2]. The *dual system methodology* was proposed in 2009 by Waters [Wat09]. With this novel and powerful proof technique, Waters proposed an IBE scheme with constant-size MPK in the standard model. The adaptive security is proven based on standard and static complexity assumptions, and the security loss is proportional to the amount of secret keys held by the adversary. This is the first IBE scheme achieving all these features simultaneously.

Since Waters deals with only one secret key at a time in the proof, a security loss of such an order of magnitude seems to be inherent. Fortunately, Chen and Wee [CW13] combined the proof idea underlying Naor-Reingold PRF [NR04] and the dual system methodology and showed an *almost-tightly* secure IBE scheme. Here *almost tight* means the security loss can be bounded by a polynomial in security parameter λ instead of the number of revealed secret keys. Soon afterwards, Blazy *et al.* [BKP14] described a generic transformation from affine MAC to IBE and constructed an affine MAC with almost-tight reduction. Their method essentially follows Chen and Wee's [CW13] but leads to a more efficient IBE. Recently, the study of almost-tightly secure IBE has been extended to the *multi-instance, multi-ciphertext setting* [HKS15,GCD+16,AHY15,GDCC16]. However the following two problems left by Chen and Wee [CW13] in 2013 still remain open.

Question 1. Can we achieve master public key of constant size?
Question 2. Can we achieve constantly tight reduction?

It's worth noting that Attrapadung *et al.* [AHY15] provided an almost-tightly secure IBE scheme achieving a trade-off between the size of master public key and sizes of secret keys and ciphertexts. As a special case, they can indeed reach constant-size master public key but at the cost of larger secret keys and ciphertexts (and vice versa). Here we do not consider this as a satisfactory solution to Chen and Wee's first open problem. One must preserve advantages of Chen and Wee's IBE such as constant-size secret keys and ciphertexts.

[1] In the selective model, the adversary has to choose the target identity ID^* before seeing MPK. Obviously, it's weaker than Boneh and Franklin's adaptive security model.

[2] In a q-type assumption, adversary's input is of size $O(q)$. The parameter q depends on the amount of secret keys revealed to adversary in Gentry's security result [Gen06].

1.1 Our Contribution

In this paper, we present an IBE scheme in the composite-order bilinear group [BGN05] with constant-size master public key, ciphertexts, and secret keys. The adaptive security in the multi-instance, multi-ciphertext setting relies on several concrete decisional subgroup assumptions [BWY11] and a subgroup variant of decisional bilinear Diffie-Hellman (DBDH) assumption. The security reduction arises a probability loss of $O(\log q)$ in which q is the upper bound of the *total* number of secret keys and challenge ciphertexts revealed to adversary in each *single* instance.

We make a comparison in Table 1. On one hand, our IBE has the shortest master public key, ciphertexts, secret keys and fastest decryption algorithm Dec. In fact the performance is nearly optimal as Wee's petit IBE [Wee16]. On the other hand, we achieve a tighter reduction in a concrete sense[3]. Under the typical setting where $q = 2^{30}$ and $n = 128$, the security loss of our IBE scheme is just a quarter of those for all previous ones [CW13,HKS15,AHY15]. Therefore our result (partially) answers Chen and Wee's first open problem and makes a significant progress on the second one. We emphasize that the multi-instance, multi-ciphertext setting [HKS15] is more realistic and complex than Boneh and Franklin's standard security notion [BF01]. This means that we are actually working on Chen and Wee's open problems in a more complex setting.

Table 1. Comparison among existing tightly secure IBE schemes in *composite*-order bilinear groups.

scheme	$\lvert\mathsf{MPK}\rvert$	$\lvert\mathsf{SK}\rvert$	$\lvert\mathsf{CT}\rvert + \lvert\mathsf{KEY}\rvert$	Dec	**tightness**	$\#p_i$	**mimc**
[CW13]	$O(n)\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$	$2\lvert\mathbb{G}\rvert$	$2\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$	$2\mathbb{P}$	$O(n)$	3	no
[HKS15]	$O(n)\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$	$2\lvert\mathbb{G}\rvert$	$2\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$	$2\mathbb{P}$	$O(n)$	4	yes
[AHY15]	$O(n)\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$	$2\lvert\mathbb{G}\rvert$	$2\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$	$2\mathbb{P}$	$O(n)$	4	yes
	$8\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$	$O(n)\lvert\mathbb{G}\rvert$	$O(n)\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$	$O(n)\mathbb{P}$	$O(n)$	4	yes
ours	$2\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$	$\lvert\mathbb{G}\rvert$	$\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert$	$1\mathbb{P}$	$O(\log q)$	4	yes

- In the table, n is the binary length of identity, q is the upper bound of total number of secret keys and challenge ciphertexts revealed to adversary in each instance.
- Let $(N, \mathbb{G}, \mathbb{G}_T, e)$ be a composite-order bilinear group. We use $\lvert\mathbb{G}\rvert, \lvert\mathbb{G}_T\rvert, \mathbb{P}$ to indicate the element size in $\mathbb{G}$ and $\mathbb{G}_T$ and the cost of one pairing operation, respectively.
- Column "$\#p_i$" shows the number of prime factors of group order N.
- Column "**mimc**" indicates whether the adaptive security can be proved in the multi-instance, multi-ciphertext setting.
- The two sub-rows of row "[AHY15]" are for scheme Φ_{cc}^{comp} and Φ_{slp}^{comp} (c.f. [AHY15] for details), respectively. Note that Φ_{slp}^{comp} employs the trade-off technique we have mentioned, and we just show the parameter of the instantiation with constant MPK in the table.

1.2 Technical Overview

Strategy. Chen and Wee [CW13] pointed out that solving the first open problem, i.e., reducing the master public key size (to a constant), may require some kinds of progresses in the underlying Naor-Reingold PRF [NR04], which is another long-standing open problem, i.e., how to reduce the seed length? As our high-level strategy, we reverse the problem in order to circumvent the

[3] Let λ be the security parameter. In the common case that $n = \mathsf{poly}(\lambda)$ and $q = \mathsf{poly}(\lambda)$, we can see that $O(n)$ and $O(\log q)$ are equivalent to $O(\lambda)$ and $O(\log \lambda)$, respectively. Superficially, our reduction is also tighter in an asymptotical sense. However $O(\log \lambda)$ here hides an adversarially-dependent constant while $O(\lambda)$ is totally independent of adversary.

technical difficulty. In particular, instead of reducing the size of master public key of a tightly secure IBE to a constant, we try to

improve the tightness of an IBE scheme already with constant-size master public key.

Technically, we propose a variant of Wee's petit IBE [Wee16] which (1) is tightly secure and (2) inherits all advantages from Wee's petit IBE. Luckily, the security loss of our variant is much smaller (in a concrete sense), which means we take a step (from almost tight) towards constantly tight reduction at the same time.

Our method is inspired by Chen and Wee's tight reduction technique from a very high level and brings Chase and Meiklejohn's idea [CM14] back to Wee's petit IBE [Wee16] in order to fulfil the intuition. We now give an overview with more technical details.

Basic Method. Assume composite-order bilinear group $(N = p_1p_2p_3, G, G_T, e)$. Let's review Wee's petit IBE [Wee16]. From a high level, Wee followed the dual system methodology [Wat09] and employed Déjà Q technique [CM14] with an extension. The IBE scheme is quite elegant as we described below.

$$\begin{aligned} \text{MPK}: &\quad g_1,\ g_1^{\alpha},\ e(g_1,u),\ H \\ \text{SK}_{\text{ID}}: &\quad u^{\frac{1}{\alpha+\text{ID}}} \cdot R_3 \\ \text{CT}_{\text{ID}}: &\quad g_1^{(\alpha+\text{ID})s},\ H(e(g_1,u)^s) \cdot M \end{aligned}$$

where $g_1, u \leftarrow G_{p_1}$, $\alpha, s \leftarrow \mathbb{Z}_N$, $R_3 \leftarrow G_{p_3}$, H is selected from a pairwise independent hash family. Here we consider G_{p_1} as normal space and G_{p_2} as semi-functional space. Subgroup G_{p_3} is used to randomize secret keys.

To prove the adaptive security, he first transformed the challenge ciphertext into the form

$$\text{CT}_{\text{ID}^*}: \quad S,\quad H(e(S, \text{SK}_{\text{ID}^*})) \cdot M$$

where $S \leftarrow G_{p_1}G_{p_2}$ and SK_{ID^*} is a secret key for target identity ID^*. The core step is to "inject" enough entropy into the semi-functional space (i.e., subgroup G_{p_2}) of SK_{ID} for each ID "touched" by adversary (including the target identity ID^* in the challenge ciphertext CT_{ID^*}). Formally, define

$$f_i(x) = \sum_{j=1}^{i} \frac{r_j}{\alpha_j + x} \in \mathbb{Z}_{p_2}$$

where $r_1, \ldots, r_i, \alpha_1, \ldots, \alpha_i \leftarrow \mathbb{Z}_{p_2}$. It can be proved that, for any $q \in \mathbb{Z}^+$, function f_q behaves like a truly random function given only q input-output pairs [Wee16,CM14] (c.f. Section 2.3). Once each secret key has been transformed into the form

$$\text{SK}_{\text{ID}}: \quad u^{\frac{1}{\alpha+\text{ID}}} \cdot \boxed{g_2^{f_q(\text{ID})}} \cdot R_3$$

where g_2 is a random generator of G_{p_2} and q depends on the total number of identities "touched" by adversary, the adaptive security will be implied by the property of random functions.

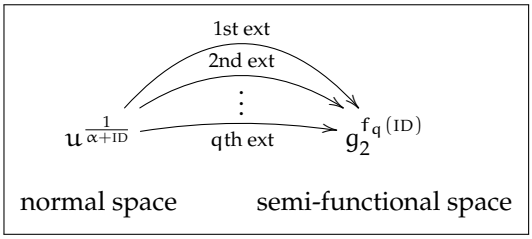
To reach the configuration, Wee transformed all involved secret keys

$$\text{from} \quad u^{\frac{1}{\alpha+\text{ID}}} \cdot \boxed{g_2^{f_0(\text{ID})}} \cdot R_3 \qquad \text{into} \quad u^{\frac{1}{\alpha+\text{ID}}} \cdot \boxed{g_2^{f_q(\text{ID})}} \cdot R_3$$

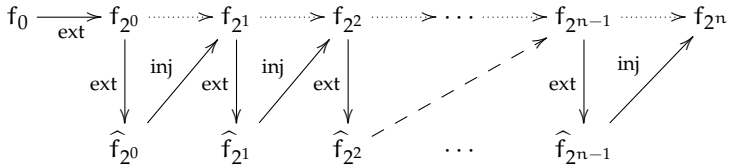
in q steps following the roadmap

$$f_0 \to f_1 \to f_2 \to \cdots \to f_q.$$

Note that $f_0(\mathrm{ID}) = 0$ for all ID. In the kth step, he extracted one unit of entropy r_k and α_k from the normal space (more accurately, from u and α) and injected them into the semi-functional space (i.e., into f_{k-1}). We illustrate the process in the graph below.

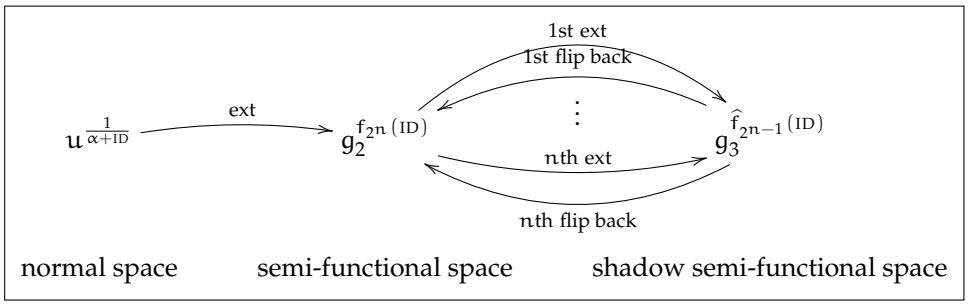


Chen and Wee's success [CW13] suggests that one must reach f_q much more quickly in order to obtain tighter reduction. To do so, we can try to extract and inject more entropy each time. Our idea is to extract entropy from f_k ($1 \leqslant k \leqslant q$) itself rather than from u and α, and then inject them back into f_k. A key observation is that f_k already has k units of entropy (i.e., $\alpha_1, r_1, \ldots, \alpha_k, r_k$) and the structure of f_k allows us to reach f_{2k} directly which will include $2k$ units of entropy. This will significantly accelerate the process towards f_q. In particular, the roadmap now becomes



where $\widehat{f}_k$ indicates the entropy extracted from f_k, both of which have the same structure but $\widehat{f}_k$ are defined by independent randomness. It's not hard to see that we only need $n = \lceil \log q \rceil + 1$ steps to reach f_q.

To fulfill the above intuition, we introduce another semi-functional space, which we call *shadow semi-functional space*, to temporarily store the entropy extracted from f_k (i.e., $\widehat{f}_k$ in the above graph) since we obviously can not put them into the normal space. Furthermore the new semi-functional space should allow us to flip all entropy back to the old semi-functional space as Chase and Meiklejohn [CM14] did. We sketch our method in the following graph where the IBE is now put into a bilinear group of order $N = p_1p_2p_3p_4$. Subgroup $\mathbb{G}_{p_3}$ acts as the shadow semi-functional space and $\mathbb{G}_{p_4}$ is used to randomize secret key.



We first extract one unit entropy from u and α and puts them into the semi-funtional space as [Wee16] which forms $f_{2^0} = f_1$. In the kth step, we first

$$\text{extract} \quad g_3^{\widehat{f}_{2k-1}(\mathrm{ID})} \quad \text{from} \quad g_2^{f_{2k-1}(\mathrm{ID})}$$

and then

$$\text{flip} \quad g_3^{\widehat{f}_{2k-1}(\mathrm{ID})} \quad \text{back as} \quad g_2^{\widehat{f}_{2k-1}(\mathrm{ID})}$$

which forms $g_2^{f_{2k}(\mathrm{ID})}$ together with $g_2^{f_{2k-1}(\mathrm{ID})}$ by defining

$$f_{2k}(\mathrm{ID}) = f_{2k-1}(\mathrm{ID}) + \widehat{f}_{2k-1}(\mathrm{ID}) \bmod p_2.$$

All these technical steps can be realized under several concrete instantiations of decisional subgroup assumption.

On the Multi-ciphertext Setting. We find that Wee's proof idea [Wee16] and our variant (see above) can be directly extended to the (single-instance) *multi-ciphertext* setting but with the restriction that only one challenge ciphertext is allowed for each target identity. This is the *weak* version of adaptive security in the multi-ciphertext setting [HKS15]. The positive aspects of the conclusion follow from two observations: (1) Each challenge ciphertext has its own randomness s which is sufficient for hiding α on the ciphertext side. That is we can always argue

$$\{g_1^{(\alpha+\mathrm{ID})s},\ e(g_1,u)^s\} = \{g_1^{(\alpha+\mathrm{ID})s},\ e(g_1^{(\alpha+\mathrm{ID})s}, u^{\frac{1}{\alpha+\mathrm{ID}}})\} = \{g_1^s,\ e(g_1^s, u^{\frac{1}{\alpha+\mathrm{ID}}})\}$$

even when there are more than one challenge ciphertexts; (2) It's adequate to cope with more than one target identity by setting $n = \lceil \log q_\sigma \rceil$ where q_σ is the total number of revealed keys and challenge ciphertexts. The restriction, which is the negative aspect of conclusion above, is set here so as to avoid the following situation: After reaching f_{2^n}, all l (with $l > 1$) challenge ciphertexts for target identity ID^* will be in the form

$$\begin{aligned}
&S_1, \quad H(e(S_1, u^{\frac{1}{\alpha+\mathrm{ID}^*}}) \cdot \boxed{e(S_1, g_2^{f_{2^n}(\mathrm{ID}^*)})}) \cdot M_1, \qquad S_1 \leftarrow \mathbb{G}_{p_1}\mathbb{G}_{p_2}\mathbb{G}_{p_3};\\
&S_2, \quad H(e(S_2, u^{\frac{1}{\alpha+\mathrm{ID}^*}}) \cdot \boxed{e(S_2, g_2^{f_{2^n}(\mathrm{ID}^*)})}) \cdot M_2, \qquad S_2 \leftarrow \mathbb{G}_{p_1}\mathbb{G}_{p_2}\mathbb{G}_{p_3};\\
&\qquad\qquad\qquad\qquad\vdots\\
&S_l, \quad H(e(S_l, u^{\frac{1}{\alpha+\mathrm{ID}^*}}) \cdot \boxed{e(S_l, g_2^{f_{2^n}(\mathrm{ID}^*)})}) \cdot M_l, \qquad S_l \leftarrow \mathbb{G}_{p_1}\mathbb{G}_{p_2}\mathbb{G}_{p_3};
\end{aligned}$$

where boxed terms have their own randomness $S_1, \ldots, S_l$, but share the same $f_{2^n}(\mathrm{ID}^*)$.

To remove this restriction and achieve the *full* adaptive security [HKS15], we employ a subgroup variant of decisional bilinear Diffie-Hellman (DBDH) assumption (in subgroup $\mathbb{G}_{p_2}$). This allows us to utilize randomness $S_1, \ldots, S_l$ and argues that the joint distribution of all boxed terms sharing $f_{2^n}(\mathrm{ID}^*)$ are pseudorandom. Our proof idea is almost the same as [HKS15] but the assumption in our case is slightly simpler.

On the Multi-instance Setting. Hofheinz *et al.* [HKS15] also investigated the so-called *multi-instance* setting where adversary is allowed to attack multiple IBE instances at the same time. Fortunately, our technique and result in the single-instance setting (see above) can be extended to the multi-instance setting with a tiny adjustment. The high-level idea is to apply our proof technique (for the single-instance setting) to each instance in an *independent* and *concurrent* way.

Assume there are τ instances. For the ι-th ($1 \leqslant \iota \leqslant \tau$) instance, we define a series of functions $f_{2^0}^{(\iota)}, \ldots, f_{2^n}^{(\iota)}$ as in the single-instance setting, which are independent of those for other instances. Here we let $n = \lceil \log \hat{q}_\sigma \rceil$ in which $\hat{q}_\sigma$ is the upper bound of the total number of revealed secret keys and challenge ciphertexts in each *single* instance. We depict the process in the graph below. In the kth step, we can create τ functions $f_{2^k}^{(1)}, \ldots, f_{2^k}^{(\tau)}$ at a time thanks to the random self-reducibility

of the decisional subgroup assumption.

$$\begin{array}{rccccccc}
\text{1st instance:} & f_{2^0}^{(1)} & & f_{2^1}^{(1)} & & f_{2^2}^{(1)} & & f_{2^n}^{(1)} \\
\text{2nd instance:} & f_{2^0}^{(2)} & \longrightarrow & f_{2^1}^{(2)} & \longrightarrow & f_{2^2}^{(2)} & \longrightarrow \cdots \longrightarrow & f_{2^n}^{(2)} \\
\vdots & & & & & & & \\
\tau\text{th instance:} & f_{2^0}^{(\tau)} & & f_{2^1}^{(\tau)} & & f_{2^2}^{(\tau)} & & f_{2^n}^{(\tau)} \\
 & & & \text{1st step} & & \text{2nd step} & & n\text{th step}
\end{array}$$

Then, utilizing the random self-reducibility of the subgroup variant of DBDH assumption, we can prove the full adaptive security in the multi-instance setting.

1.3 Related Work

The dual system methodology has been successfully applied to broader area of functional encryptions [OT10,LOS+10]. In 2014, Wee [Wee14] and Attrapadung [Att14] independently gave generic constructions of a large class of functional encryptions with adaptive security including attribute based encryption, inner-product encryption, and even functional encryption for regular language. They introduced the notion of predicate/pair encoding and employed the dual system methodology in the composite-order bilinear group. Their results have been extended to the prime-order groups [AC16,Att16,CGW15] recently.

Tight reduction under short public parameter has been studied in the field of digital signature. Very recently, Hofheinz developed algebraic partitioning technique [Hof16b] and adaptive partitioning technique [Hof16a] based on Chen and Wee's result [CW13], which leaded to tightly secure signatures with constant verification key and public key encryption against chosen ciphertext attack with similar features. However it's not quite direct to apply his techniques to IBE.

Déjà Q technique was proposed by Chase and Meiklejohn [CM14]. They showed that one can avoid the use of (a class of) q-type assumptions with the help of a composite-order bilinear group equipped with decisional subgroup assumption using the dual system methodology. Recently, Wee gave a petit IBE scheme and broadcast encryption scheme [Wee16] with an extended Déjà Q technique. Their results have been used to build non-zero inner-product encryptions [CLR16] and functional commitments for linear functions [LRY16] (which imply many other important primitives such as accumulators.)

A recent work by Boyen and Li [BL16] established a generic framework from PRF to signatures and IBE utilizing the powerful tools in the lattice world. The reduction is constantly tight and the security loss of resulting scheme solely depends on that of underlying PRF. We remark that all tightly secure IBE schemes they showed still require non-constant-size master public key.

Independent Work. An independent work by Chase, Maller and Meiklejohn [CMM16] developed the basic Déjà Q technique [CM14] in a similar way to us. We focus on solving or making progress on open problems left by Chen and Wee [CW13] in a specific area (i.e., tightly secure IBE) while Chase *et al.* focus on a more general goal, i.e., tightly translating a broader class of q-type assumptions into static ones. Although they described four functional encryptions including an IBE scheme, its master public key consists of $O(n)$ group elements with identity space $\{0,1\}^n$. As a matter of fact, neither Wee's petit IBE [Wee16] nor ours can be derived from an IBE under q-type assumptions using Chase *et al.*'s new framework [CMM16]. Therefore we believe it's still necessary to propose and analyze the IBE directly.

Open Problem. Our proposed IBE scheme works in the composite-order bilinear group which can be a drawback from a practical viewpoint. We leave it as an open problem to find a prime-order IBE with tight(er) reduction, constant-size master public key, secret keys and ciphertexts.

In fact, to our best knowledge, no result based on Déjà Q technique has been adapted to the prime-order group.

Organization. The paper will be organized as follows. Section 2 reviews several basic notions, the decisional subgroup assumption and a core lemma given by Wee [Wee16]. Section 3 describes our IBE scheme and proves the *weak* adaptive security in the single-instance, multi-ciphertext setting. We then show how to extend the basic result to *full* adaptive security and *multi-instance* setting in Section 4 and Section 5, respectively.

2 Preliminaries

Notation. Let S be a finite set. The notation $s \leftarrow S$ means that we pick s from S at random. "p.p.t." is the abbreviation of "probabilistic polynomial time".

2.1 Composite-order Bilinear Groups

Our IBE scheme is constructed in composite-order bilinear groups [BGN05]. We assume a group generator GrpGen which takes as input the security parameter 1^λ and outputs group description $\mathcal{G} = (N, \mathbb{G}, \mathbb{G}_T, e)$, where order N is product of 4 distinct $\Theta(\lambda)$-bit primes, group $\mathbb{G}$ and $\mathbb{G}_T$ are all finite cyclic groups of order N and e is an efficient, non-degenerated bilinear map from $\mathbb{G} \times \mathbb{G}$ to $\mathbb{G}_T$. With $N = p_1p_2p_3p_4$ for primes p_1, p_2, p_3, p_4, we let $\mathbb{G}_{p_i}$ be the subgroup of order p_i in $\mathbb{G}$ and use $\mathbb{G}_{p_i}^*$ to refer to the set of all generators in $\mathbb{G}_{p_i}$, i.e, $\mathbb{G}_{p_i} \setminus \{1\}$.

We review several concrete instantiations of decisional subgroup assumption [BWY11]. Since we can uniquely decompose $\mathbb{G} = \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3} \times \mathbb{G}_{p_4}$, we employ a special notation for sampling random elements from a composite-order *subgroup* of $\mathbb{G}$. For any two prime factors p_i, p_j of N with $1 \leqslant i < j \leqslant 4$, we use $X_iX_j \leftarrow \mathbb{G}_{p_i}\mathbb{G}_{p_j}$ to indicate that we uniformly sample an element from the subgroup of order p_ip_j, whose respective components in $\mathbb{G}_{p_i}$ and $\mathbb{G}_{p_j}$ are X_i and X_j. The notation can be naturally extended to any subgroups.

Assumption 1 (SD1) *For any p.p.t. adversary $\mathcal{A}$ the following advantage function is negligible in λ.*

$$\mathsf{Adv}^{\mathrm{SD1}}_{\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(\mathcal{G}, g_1, g_4, T_0) = 1] - \Pr[\mathcal{A}(\mathcal{G}, g_1, g_4, T_1)]|,$$

where $\mathcal{G} \leftarrow \mathsf{GrpGen}(1^\lambda)$, $g_1 \leftarrow \mathbb{G}_{p_1}^*$, $g_4 \leftarrow \mathbb{G}_{p_4}^*$,

$$T_0 \leftarrow \mathbb{G}_{p_1} \quad \textit{and} \quad T_1 \leftarrow \mathbb{G}_{p_1}\mathbb{G}_{p_2}\mathbb{G}_{p_3}.$$

Assumption 2 (SD2) *For any p.p.t. adversary $\mathcal{A}$ the following advantage function is negligible in λ.*

$$\mathsf{Adv}^{\mathrm{SD2}}_{\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(\mathcal{G}, g_1, g_4, X_1X_2X_3, T_0) = 1] - \Pr[\mathcal{A}(\mathcal{G}, g_1, g_4, X_1X_2X_3, T_1)]|,$$

where $\mathcal{G} \leftarrow \mathsf{GrpGen}(1^\lambda)$, $g_1 \leftarrow \mathbb{G}_{p_1}^*$, $g_4 \leftarrow \mathbb{G}_{p_4}^*$, $X_1X_2X_3 \leftarrow \mathbb{G}_{p_1}\mathbb{G}_{p_2}\mathbb{G}_{p_3}$,

$$T_0 \leftarrow \mathbb{G}_{p_1} \quad \textit{and} \quad T_1 \leftarrow \mathbb{G}_{p_1}\mathbb{G}_{p_2}.$$

Assumption 3 (SD3) *For any p.p.t. adversary $\mathcal{A}$ the following advantage function is negligible in λ.*

$$\mathsf{Adv}^{\mathrm{SD3}}_{\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(\mathcal{G}, g_1, g_4, X_1X_2X_3, T_0) = 1] - \Pr[\mathcal{A}(\mathcal{G}, g_1, g_4, X_1X_2X_3, T_1)]|,$$

where $\mathcal{G} \leftarrow \mathsf{GrpGen}(1^\lambda)$, $g_1 \leftarrow \mathbb{G}_{p_1}^*$, $g_4 \leftarrow \mathbb{G}_{p_4}^*$, $X_1X_2X_3 \leftarrow \mathbb{G}_{p_1}\mathbb{G}_{p_2}\mathbb{G}_{p_3}$,

$$T_0 \leftarrow \mathbb{G}_{p_2} \quad \textit{and} \quad T_1 \leftarrow \mathbb{G}_{p_2}\mathbb{G}_{p_3}.$$

Assumption 4 (SD4) *For any p.p.t. adversary $\mathcal{A}$ the following advantage function is negligible in λ.*

$$\mathsf{Adv}^{\mathrm{SD4}}_{\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(\mathcal{G}, g_1, g_4, X_1X_2X_3, Y_2Y_4, T_0) = 1] - \Pr[\mathcal{A}(\mathcal{G}, g_1, g_4, X_1X_2X_3, Y_2Y_4, T_1)]|,$$

where $\mathcal{G} \leftarrow \mathsf{GrpGen}(1^\lambda)$, $g_1 \leftarrow \mathbb{G}_{p_1}^*$, $g_4 \leftarrow \mathbb{G}_{p_4}^*$, $X_1X_2X_3 \leftarrow \mathbb{G}_{p_1}\mathbb{G}_{p_2}\mathbb{G}_{p_3}$, $Y_2Y_4 \leftarrow \mathbb{G}_{p_2}\mathbb{G}_{p_4}$,

$$T_0 \leftarrow \mathbb{G}_{p_2}\mathbb{G}_{p_4} \quad \textit{and} \quad T_1 \leftarrow \mathbb{G}_{p_3}\mathbb{G}_{p_4}.$$

2.2 Identity Based Encryptions

In the paper we define the notion of identity based encryption (IBE) in the framework of key encapsulation mechanism (KEM).

Algorithms. An IBE (in the single-instance setting) is composed of the following four p.p.t. algorithms:

- $\mathsf{Setup}(1^\lambda) \to (\mathrm{MPK}, \mathrm{MSK})$. The *setup algorithm* Setup takes as input the security parameter 1^λ and outputs master public/secret key pair $(\mathrm{MPK}, \mathrm{MSK})$. We assume that MPK includes ciphertext space $\mathcal{C}$ and key space $\mathcal{K}$.
- $\mathsf{KeyGen}(\mathrm{MPK}, \mathrm{MSK}, \mathrm{ID}) \to \mathrm{SK}$. The *key generation algorithm* KeyGen takes as input the master public key MPK, the master secret key MSK and an identity ID and outputs its secret key SK.
- $\mathsf{Enc}(\mathrm{MPK}, \mathrm{ID}) \to (\mathrm{CT}, \mathrm{KEY})$. The *encryption algorithm* Enc takes as input the master public key MPK and an identity ID and outputs a ciphertext $\mathrm{CT} \in \mathcal{C}$ along with key $\mathrm{KEY} \in \mathcal{K}$.
- $\mathsf{Dec}(\mathrm{MPK}, \mathrm{CT}, \mathrm{SK}) \to \mathrm{KEY}$. The *decryption algorithm* Dec takes as input the master public key MPK, a ciphertext CT and a secret key SK and outputs key KEY or $\perp$.

Correctness. For any $\lambda \in \mathbb{N}$, $(\mathrm{MPK}, \mathrm{MSK}) \in [\mathsf{Setup}(1^\lambda)]$, identity ID, we require

$$\Pr\left[\mathsf{Dec}(\mathrm{MPK}, \mathrm{CT}, \mathrm{SK}) = \mathrm{KEY} \;\middle|\; \begin{array}{c} \mathrm{SK} \leftarrow \mathsf{KeyGen}(\mathrm{MPK}, \mathrm{MSK}, \mathrm{ID}) \\ (\mathrm{CT}, \mathrm{KEY}) \leftarrow \mathsf{Enc}(\mathrm{MPK}, \mathrm{ID}) \end{array}\right] \geqslant 1 - 2^{-\Omega(\lambda)}.$$

The probability space is defined by random coins of KeyGen and Enc.

Security notion. For any adversary $\mathcal{A}$, we define the advantage function as

$$\mathsf{Adv}^{\mathrm{IBE}}_{\mathcal{A}}(\lambda) = \left|\Pr\left[\beta = \beta' \;\middle|\; \begin{array}{c} (\mathrm{MPK}, \mathrm{MSK}) \leftarrow \mathsf{Setup}(1^\lambda),\ \beta \leftarrow \{0,1\} \\ \beta' \leftarrow \mathcal{A}^{\mathsf{O}^{\mathsf{KeyGen}}(\cdot), \mathsf{O}^{\mathsf{Enc}}_\beta(\cdot)}(1^\lambda, \mathrm{MPK}) \end{array}\right] - \frac{1}{2}\right|$$

where oracles are defined as

- $\mathsf{O}^{\mathsf{KeyGen}}$: On input (ID), the oracle returns $\mathsf{KeyGen}(\mathrm{MPK}, \mathrm{MSK}, \mathrm{ID})$ and sets $Q_K = Q_K \cup \{\mathrm{ID}\}$.
- $\mathsf{O}^{\mathsf{Enc}}_\beta$: On input (ID^*), the oracle samples $(\mathrm{CT}^*_1, \mathrm{KEY}^*_1) \leftarrow \mathsf{Enc}(\mathrm{MPK}, \mathrm{ID}^*)$, $(\mathrm{CT}^*_0, \mathrm{KEY}^*_0) \leftarrow \mathcal{C} \times \mathcal{K}$ and returns $(\mathrm{CT}^*_\beta, \mathrm{KEY}^*_\beta)$. It then sets $Q_C = Q_C \cup \{\mathrm{ID}^*\}$.

The probability is defined over random coins used by Setup, oracle $\mathsf{O}^{\mathsf{KeyGen}}$ and $\mathsf{O}^{\mathsf{Enc}}_\beta$, and adversary $\mathcal{A}$ as well as random bit β. We say an IBE is *adaptively secure and anonymous* if and only if the above advantage function is negligible in λ for any p.p.t. adversary such that $Q_C \cap Q_K = \emptyset$.

2.3 A Core Lemma

We review the lemma by Wee [Wee16] as follows.

Lemma 1. *Fix a prime* p. *For any adversary* $\mathcal{A}$ *making at most* q *queries, we have*

$$\left|\Pr\left[\mathcal{A}^{\mathsf{O}^f(\cdot)}(1^q) = 1\right] - \Pr\left[\mathcal{A}^{\mathsf{O}^{\mathsf{RF}}(\cdot)}(1^q) = 1\right]\right| \leqslant \frac{q^2}{p}$$

where oracles are defined as

- O^f: *The oracle is initialized by picking* $r_1, \ldots, r_q, \alpha_1, \ldots, \alpha_q \leftarrow \mathbb{Z}_p$. *On input* $x \in \mathbb{Z}_p$, *it outputs*

$$f_{r_1,\ldots,r_q,\alpha_1,\ldots,\alpha_q}(x) = \sum_{i=1}^{q} \frac{r_i}{\alpha_i + x} \in \mathbb{Z}_p.$$

  Every queries are answered using the same $r_1, \ldots, r_q, \alpha_1, \ldots, \alpha_q$ *we picked at the very beginning.*
- O^{RF}: *This oracle behaves as a truly random function* $\mathsf{RF} : \mathbb{Z}_p \to \mathbb{Z}_p$. *On input* $x \in \mathbb{Z}_p$, *it returns* $\mathsf{RF}(x)$ *if it has been defined, otherwise it returns* $y \leftarrow \mathbb{Z}_p$ *and defines* $\mathsf{RF}(x) = y$.

3 Our Basic IBE Scheme

This section describes our IBE scheme. At current stage, we prove its *weak* adaptive security and anonymity in the *single-instance*, multi-challenge setting, i.e., adversary can access only one IBE instance and only one challenge ciphertext is allowed for *each* target identity.

3.1 Construction

Our IBE scheme is described as follows.

– $\mathsf{Setup}(1^\lambda)$. Run $\mathcal{G} = (N, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathsf{GrpGen}(1^\lambda)$. Sample

$$\alpha \leftarrow \mathbb{Z}_N, \quad g_1 \leftarrow \mathbb{G}_{p_1}^*, \quad u \leftarrow \mathbb{G}_{p_1}, \quad g_4 \leftarrow \mathbb{G}_{p_4}^*.$$

Pick $H : \mathbb{G}_T \to \{0,1\}^\lambda$ from a pairwise independent hash family. Output

$$\mathrm{MPK} = (g_1,\ g_1^\alpha,\ e(g_1, u),\ H) \quad \text{and} \quad \mathrm{MSK} = (\alpha, u, g_4).$$

– $\mathsf{KeyGen}(\mathrm{MPK}, \mathrm{MSK}, \mathrm{ID})$. Sample $R_4 \leftarrow \mathbb{G}_{p_4}$ and output

$$\mathrm{SK} = u^{\frac{1}{\alpha+\mathrm{ID}}} \cdot R_4.$$

– $\mathsf{Enc}(\mathrm{MPK}, \mathrm{ID})$. Sample $s \leftarrow \mathbb{Z}_N$ and output

$$\mathrm{CT} = g_1^{(\alpha+\mathrm{ID})s} \quad \text{and} \quad \mathrm{KEY} = H(e(g_1, u)^s).$$

– $\mathsf{Dec}(\mathrm{MPK}, \mathrm{CT}, \mathrm{SK})$. Return

$$\mathrm{KEY} = H(e(\mathrm{CT}, \mathrm{SK})).$$

Correctness. We have

$$e(\mathrm{CT}, \mathrm{SK}) = e(g_1^{(\alpha+\mathrm{ID})s}, u^{\frac{1}{\alpha+\mathrm{ID}}} \cdot R_4) = e(g_1, u)^{(\alpha+\mathrm{ID})s \cdot \frac{1}{\alpha+\mathrm{ID}}} = e(g_1, u)^s.$$

This immediately proves the correctness.

3.2 Security Analysis: An Overview

We prove the following theorem.

Theorem 1. *For any p.p.t. adversary $\mathcal{A}$ sending at most q_σ queries to O^{KeyGen} and O_β^{Enc}, there exist $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$, $\mathcal{B}_4$ such that*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathrm{IBE}}(\lambda) \leqslant \tfrac{5}{2} \cdot \mathsf{Adv}_{\mathcal{B}_1}^{\mathrm{SD1}}(\lambda) + 2 \cdot \mathsf{Adv}_{\mathcal{B}_2}^{\mathrm{SD2}}(\lambda) + 2 \cdot \lceil \log q_\sigma \rceil \cdot \mathsf{Adv}_{\mathcal{B}_3}^{\mathrm{SD3}}(\lambda) + \\ + \big(2 \cdot \lceil \log q_\sigma \rceil + \tfrac{1}{2}\big) \cdot \mathsf{Adv}_{\mathcal{B}_4}^{\mathrm{SD4}}(\lambda) + 2^{-\Omega(\lambda)}$$

and $\max\{T(\mathcal{B}_1), T(\mathcal{B}_2), T(\mathcal{B}_3), T(\mathcal{B}_4)\} \approx T(\mathcal{A}) + q_\sigma^2 \cdot \mathsf{poly}(\lambda)$.

We prove the theorem using hybrid argument. We define the advantage function of any p.p.t. adversary $\mathcal{A}$ in Game_{xxx} as

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_{xxx}}(\lambda) = |\Pr[\beta = \beta'] - 1/2|.$$

Let $n = \lceil \log q_\sigma \rceil$. Our proof employs the following game sequence.

$\mathsf{Game}_{\mathsf{real}}$ is the real game.

Game_0 is the real game with the following assumptions:

- $\mathcal{A}$ can not find $\mathrm{ID}, \mathrm{ID}' \in \mathbb{Z}_N$ such that $\mathrm{ID} \neq \mathrm{ID}'$ but $\mathrm{ID} = \mathrm{ID}' \mod p_2$;
- $\mathcal{A}$ can not find $\mathrm{ID} \in \mathbb{Z}_N$ such that $\alpha + \mathrm{ID} = 0 \mod p_1$.

One may notice that $\mathcal{A}$ can efficiently factorize the order N and break the general decisional subgroup assumption when it violates one of the above two assumptions. Technically, Game_0 aborts immediately when $\mathcal{A}$ submits $\mathrm{ID} \in \mathbb{Z}_N$ (through $\mathsf{O}^{\mathsf{KeyGen}}$ or $\mathsf{O}^{\mathsf{Enc}}_\beta$) such that

- $\gcd(\mathrm{ID} - \mathrm{ID}', N) \notin \{1, N\}$ for some previous identity $\mathrm{ID}' \in \mathbb{Z}_N$;
- $\gcd(\alpha + \mathrm{ID}, N) \notin \{1, N\}$.

Note that both $N \in \mathbb{Z}$ and $\alpha \in \mathbb{Z}_N$ are always available throughout our proof. We prove the following lemma.

Lemma 2 (from $\mathsf{Game}_{\mathsf{real}}$ to Game_0). *For any p.p.t. adversary $\mathcal{A}$ sending at most q_σ queries to $\mathsf{O}^{\mathsf{KeyGen}}$ and $\mathsf{O}^{\mathsf{Enc}}_\beta$, there exist $\mathcal{B}_1, \mathcal{B}_2$ such that $\max\{\mathsf{T}(\mathcal{B}_1), \mathsf{T}(\mathcal{B}_2)\} \approx \mathsf{T}(\mathcal{A}) + q_\sigma \cdot \mathsf{poly}(\lambda)$ and*

$$|\mathsf{Adv}^{\mathsf{Game}_0}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{Game}_{\mathsf{real}}}_{\mathcal{A}}(\lambda)| \leqslant \frac{1}{2} \cdot \mathsf{Adv}^{\mathsf{SD1}}_{\mathcal{B}_1}(\lambda) + \frac{1}{2} \cdot \mathsf{Adv}^{\mathsf{SD4}}_{\mathcal{B}_2}(\lambda) + 2^{-\Omega(\lambda)}.$$

Game'_0 is identical to Game_0 except that, for each query (ID^*) to $\mathsf{O}^{\mathsf{Enc}}_\beta$, we compute KEY^*_1 as

$$\mathrm{KEY}^*_1 = \mathsf{H}(e(\mathrm{CT}^*_1, \mathrm{SK}_{\mathrm{ID}^*}))$$

where CT^*_1 is produced as before and $\mathrm{SK}_{\mathrm{ID}^*}$ is obtained via a $\mathsf{O}^{\mathsf{KeyGen}}$ query (ID^*). From the correctness, we have that

$$\mathsf{Adv}^{\mathsf{Game}'_0}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathsf{Game}_0}_{\mathcal{A}}(\lambda)$$

for any p.p.t. adversary $\mathcal{A}$.

Game''_0 is identical to Game'_0 except that, for each query (ID^*) to $\mathsf{O}^{\mathsf{Enc}}_\beta$, we compute CT^*_1 as

$$g_1^s \qquad \text{instead of} \qquad g_1^{(\alpha + \mathrm{ID}^*)s}$$

where $s \leftarrow \mathbb{Z}_N$. We have

$$\mathsf{Adv}^{\mathsf{Game}''_0}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathsf{Game}'_0}_{\mathcal{A}}(\lambda)$$

for any p.p.t. adversary $\mathcal{A}$ since the two games are exactly the same unless $\alpha + \mathrm{ID}^* = 0 \bmod p_1$ for some query (ID^*). We emphasize that it holds even for the multiple challenge setting since s is freshly picked for each query.

Game_1 is identical to Game''_0 except that, for each query (ID^*) to $\mathsf{O}^{\mathsf{Enc}}_\beta$, we compute CT^*_1 as

$$(g_1 g_2 g_3)^s \qquad \text{instead of} \qquad g_1^s$$

where $s \leftarrow \mathbb{Z}_N$, $g_2 \leftarrow \mathbb{G}^*_{p_2}$ and $g_3 \leftarrow \mathbb{G}^*_{p_3}$. We prove the lemma.

Lemma 3 (from Game''_0 to Game_1). *For any p.p.t. adversary $\mathcal{A}$ sending at most q_σ queries to $\mathsf{O}^{\mathsf{KeyGen}}$ and $\mathsf{O}^{\mathsf{Enc}}_\beta$, there exists $\mathcal{B}$ with $\mathsf{T}(\mathcal{B}) \approx \mathsf{T}(\mathcal{A}) + q_\sigma \cdot \mathsf{poly}(\lambda)$ and*

$$|\mathsf{Adv}^{\mathsf{Game}_1}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{Game}''_0}_{\mathcal{A}}(\lambda)| \leqslant \mathsf{Adv}^{\mathsf{SD1}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

$\mathsf{Game}_{2.i}$ **($0 \leqslant i \leqslant n$, $n = \lceil \log q_\sigma \rceil$)** is identical to Game_1 except that, for each query (ID) to $\mathsf{O}^{\mathsf{KeyGen}}$ (including those involved in $\mathsf{O}^{\mathsf{Enc}}_\beta$), we return

$$u^{\frac{1}{\alpha + \mathrm{ID}}} \cdot \boxed{g_2^{\sum_{j=1}^{2^i} \frac{r_j}{\alpha_j + \mathrm{ID}}}} \cdot R_4$$

where $g_2 \leftarrow \mathbb{G}^*_{p_2}$ and $\alpha_j, r_j \leftarrow \mathbb{Z}_N$ for all $j \in [2^i]$. We must prove the following lemma first.

Lemma 4 (from Game_1 to $\mathsf{Game}_{2.0}$). *For any p.p.t. adversary $\mathcal{A}$ sending at most q_σ queries to $\mathsf{O}^{\mathsf{KeyGen}}$ and $\mathsf{O}^{\mathsf{Enc}}_\beta$, there exists $\mathcal{B}$ with $\mathsf{T}(\mathcal{B}) \approx \mathsf{T}(\mathcal{A}) + q_\sigma \cdot \mathsf{poly}(\lambda)$ and*

$$|\mathsf{Adv}^{\mathsf{Game}_{2.0}}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{Game}_1}_{\mathcal{A}}(\lambda)| \leqslant \mathsf{Adv}^{\mathsf{SD2}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

To move from $\mathsf{Game}_{2.i}$ to $\mathsf{Game}_{2.(i+1)}$, we need two additional games:

- $\underline{\mathsf{Game}_{2.i.1}}$ is identical to $\mathsf{Game}_{2.i}$ except that, for each query (ID) to $\mathsf{O}^{\mathsf{KeyGen}}$, we return

$$u^{\frac{1}{\alpha+\mathrm{ID}}} \cdot g_2^{\sum_{j=1}^{2^i} \frac{r_j}{\alpha_j+\mathrm{ID}}} \cdot \boxed{g_3^{\sum_{j=1}^{2^i} \frac{\widehat{r}_j}{\widehat{\alpha}_j+\mathrm{ID}}}} \cdot R_4$$

  where $g_3 \leftarrow \mathbb{G}^*_{p_3}$ and $\alpha_j, r_j, \widehat{\alpha}_j, \widehat{r}_j \leftarrow \mathbb{Z}_N$ for all $j \in [2^i]$.
- $\underline{\mathsf{Game}_{2.i.2}}$ is identical to $\mathsf{Game}_{2.i}$ except that, for each query (ID) to $\mathsf{O}^{\mathsf{KeyGen}}$, we return

$$u^{\frac{1}{\alpha+\mathrm{ID}}} \cdot g_2^{\sum_{j=1}^{2^i} \frac{r_j}{\alpha_j+\mathrm{ID}} + \boxed{\sum_{j=1}^{2^i} \frac{\widehat{r}_j}{\widehat{\alpha}_j+\mathrm{ID}}}} \cdot R_4$$

  where $\alpha_j, r_j, \widehat{\alpha}_j, \widehat{r}_j \leftarrow \mathbb{Z}_N$ for all $j \in [2^i]$.

We prove the following two lemmas for all $i \in [0, n]$.

Lemma 5 (from $\mathsf{Game}_{2.i}$ to $\mathsf{Game}_{2.i.1}$). *For any p.p.t. adversary $\mathcal{A}$ sending at most q_σ queries to $\mathsf{O}^{\mathsf{KeyGen}}$ and $\mathsf{O}^{\mathsf{Enc}}_\beta$, there exists $\mathcal{B}$ with $\mathsf{T}(\mathcal{B}) \approx \mathsf{T}(\mathcal{A}) + q_\sigma^2 \cdot \mathsf{poly}(\lambda)$ and*

$$|\mathsf{Adv}^{\mathsf{Game}_{2.i.1}}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{Game}_{2.i}}_{\mathcal{A}}(\lambda)| \leqslant \mathsf{Adv}^{\mathsf{SD3}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Lemma 6 (from $\mathsf{Game}_{2.i.1}$ to $\mathsf{Game}_{2.i.2}$). *For any p.p.t. adversary $\mathcal{A}$ sending at most q_σ queries to $\mathsf{O}^{\mathsf{KeyGen}}$ and $\mathsf{O}^{\mathsf{Enc}}_\beta$, there exists $\mathcal{B}$ with $\mathsf{T}(\mathcal{B}) \approx \mathsf{T}(\mathcal{A}) + q_\sigma^2 \cdot \mathsf{poly}(\lambda)$ and*

$$|\mathsf{Adv}^{\mathsf{Game}_{2.i.2}}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{Game}_{2.i.1}}_{\mathcal{A}}(\lambda)| \leqslant \mathsf{Adv}^{\mathsf{SD4}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Observe that all $\alpha_j, r_j, \widehat{\alpha}_j, \widehat{r}_j \in \mathbb{Z}_N$ are i.i.d. variables in $\mathsf{Game}_{2.i.2}$. By setting $\alpha_{2^i+k} = \widehat{\alpha}_k$ and $r_{2^i+k} = \widehat{r}_k$ for all $k \in [2^i]$, one can claim that

$$\mathsf{Adv}^{\mathsf{Game}_{2.i.2}}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathsf{Game}_{2.(i+1)}}_{\mathcal{A}}(\lambda)$$

for any adversary $\mathcal{A}$.

Game_3 is identical to $\mathsf{Game}_{2.n}$ except that, for each query (ID) to $\mathsf{O}^{\mathsf{KeyGen}}$, we return

$$u^{\frac{1}{\alpha+\mathrm{ID}}} \cdot g_2^{\mathsf{RF}(\mathrm{ID})} \cdot R_4$$

where $g_2 \leftarrow \mathbb{G}^*_{p_2}$ and RF is a truly random function. By Lemma 1 (i.e., core lemma), we have

$$|\mathsf{Adv}^{\mathsf{Game}_{2.n}}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{Game}_3}_{\mathcal{A}}(\lambda)| \leqslant 2^{-\Omega(\lambda)}$$

for any adversary $\mathcal{A}$.

Game_4 is identical to Game_3 except that, for each query (ID^*) to $\mathsf{O}^{\mathsf{Enc}}_\beta$, we directly sample $\mathrm{KEY}^*_1 \leftarrow \{0,1\}^\lambda$. In Game_3, we compute a challenge for ID^* as follows:

$$\mathrm{CT}^*_1 = (g_1 g_2 g_3)^s \quad \text{and} \quad \mathrm{KEY}^*_1 = \mathsf{H}(e(g_1^s, u^{\frac{1}{\alpha+\mathrm{ID}^*}}) \cdot \boxed{e(g_2, g_2)^{s \cdot \mathsf{RF}(\mathrm{ID}^*)}}).$$

Due to the restrictions in the security game, $\mathsf{RF}(\mathrm{ID}^*)$ will be evaluated only in this place and the boxed term has entropy of $p_2 = \Theta(\lambda)$ which means we can sample $\mathrm{KEY}^*_1 \leftarrow \{0,1\}^\lambda$ instead but with small error. This comes from the leftover hash lemma and the fact that the pairwise independent hash family is a stronger extractor. Formally we have

$$|\mathsf{Adv}^{\mathsf{Game}_4}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{Game}_3}_{\mathcal{A}}(\lambda)| \leqslant 2^{-\Omega(\lambda)}$$

for any adversary $\mathcal{A}$.

Utilizing, in a reversed manner, a game sequence which is identical to the above except that we always sample $\mathrm{KEY}_1^* \leftarrow \{0,1\}^\lambda$ when answering queries to $\mathsf{O}_\beta^{\mathsf{Enc}}$, we may reach a game where we create

$$\boxed{\mathrm{CT}_1^* \leftarrow \mathbb{G}_{p_1}} \quad \text{and} \quad \mathrm{KEY}_1^* \leftarrow \{0,1\}^\lambda \quad \text{for all } \mathrm{ID}^*.$$

This means we can answer all queries to $\mathsf{O}_\beta^{\mathsf{Enc}}$ without β and readily prove the main theorem.

3.3 Security Analysis: Proving All Lemmas

This subsection provides all omitted proofs.

Proof of Lemma 2.

Proof (a sketch). Let $\mathsf{Abort}_{\mathcal{A}}$ be the event that Game_0 aborts with adversary $\mathcal{A}$. We have

$$|\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_0}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_{\mathsf{real}}}(\lambda)| \leqslant \Pr[\mathsf{Abort}_{\mathcal{A}}].$$

As we have discussed, when $\mathsf{Abort}_{\mathcal{A}}$ occurs, one can reach a non-trivial factorization of N. That is we can efficiently compute $N_1, N_2 \in \mathbb{Z}$ such that $N = N_1 N_2$ and $1 < N_1, N_2 < N$. Let us consider the following three cases:

1. If $p_4 | N_1$ and $p_2 \nmid N_1$, given $(\mathcal{G}, g_1, g_4, X_1X_2X_3, Y_2Y_4, T)$ where either $T \leftarrow \mathbb{G}_{p_2}\mathbb{G}_{p_4}$ or $T \leftarrow \mathbb{G}_{p_3}\mathbb{G}_{p_4}$, we observe that $(Y_2Y_4)^{N_1} \in \mathbb{G}_{p_2}$. This allows us to break SD4 assumption by checking whether $e((Y_2Y_4)^{N_1}, T) = 1$.
2. If $p_2p_4 | N_1$ and $p_3 \nmid N_1$, given $(\mathcal{G}, g_1, g_4, X_1X_2X_3, Y_2Y_4, T)$ where either $T \leftarrow \mathbb{G}_{p_2}\mathbb{G}_{p_4}$ or $T \leftarrow \mathbb{G}_{p_3}\mathbb{G}_{p_4}$, we can break SD4 assumption by checking whether $T^{N_1} = 1$.
3. If $p_2p_3p_4 | N_1$, it must be the case that $N_2 = p_1$. Given $(\mathcal{G}, g_1, g_4, T)$ where either $T \leftarrow \mathbb{G}_{p_1}$ or $T \leftarrow \mathbb{G}_{p_1}\mathbb{G}_{p_2}\mathbb{G}_{p_3}$, we can break SD1 assumption by checking whether $T^{N_2} = 1$.

In all three cases, we have access to $(\mathcal{G}, g_1, g_4)$ which is sufficient for simulating Game_0 for $\mathcal{A}$. Therefore we can claim that there exist $\mathcal{B}_1, \mathcal{B}_2$ such that

$$\Pr[\mathsf{Abort}_{\mathcal{A}}] \leqslant \frac{1}{2} \cdot \mathsf{Adv}_{\mathcal{B}_1}^{\mathrm{SD1}}(\lambda) + \frac{1}{2} \cdot \mathsf{Adv}_{\mathcal{B}_2}^{\mathrm{SD4}}(\lambda) + 2^{-\Omega(\lambda)}$$

and $\max\{\mathsf{T}(\mathcal{B}_1), \mathsf{T}(\mathcal{B}_2)\} \approx \mathsf{T}(\mathcal{A}) + q_\sigma \cdot \mathsf{poly}(\lambda)$. This proves the lemma. □

Proof of Lemma 3.

Proof. Given $(\mathcal{G}, g_1, g_4, T)$ where either $T \leftarrow \mathbb{G}_{p_1}$ or $T \leftarrow \mathbb{G}_{p_1}\mathbb{G}_{p_2}\mathbb{G}_{p_3}$, $\mathcal{B}$ works as follows:

Initialization. Pick $\alpha \leftarrow \mathbb{Z}_N$ and $u \leftarrow \mathbb{G}_{p_1}$. Select hash function H. Output

$$\mathrm{MPK} = (g_1,\ g_1^\alpha,\ e(g_1, u),\ H)$$

and store $\mathrm{MSK} = (\alpha,\ u,\ g_4)$.

Answering $\mathsf{O}^{\mathsf{KeyGen}}$. On input (ID), return $\mathsf{KeyGen}(\mathrm{MPK}, \mathrm{MSK}, \mathrm{ID})$ directly.

Answering $\mathsf{O}_\beta^{\mathsf{Enc}}$. On input (ID^*), obtain $\mathrm{SK}_{\mathrm{ID}^*}$ via a query (ID^*) to $\mathsf{O}^{\mathsf{KeyGen}}$. Sample $s' \leftarrow \mathbb{Z}_N$ and compute

$$\mathrm{CT}_1^* = T^{s'} \quad \text{and} \quad \mathrm{KEY}_1^* = H(e(T^{s'}, \mathrm{SK}_{\mathrm{ID}^*})).$$

$\mathcal{B}$ then picks $(\mathrm{CT}_0^*, \mathrm{KEY}_0^*) \leftarrow \mathbb{G}_{p_1} \times \{0,1\}^\lambda$ and returns $(\mathrm{CT}_\beta^*, \mathrm{KEY}_\beta^*)$.

Finalize. $\mathcal{B}$ returns 1 if $\beta = \beta'$ and returns 0 in the other case.

When $T \leftarrow \mathbb{G}_{p_1}$, the simulation is identical to Game_0''; when $T \leftarrow \mathbb{G}_{p_1}\mathbb{G}_{p_2}\mathbb{G}_{p_3}$, the simulation is identical to Game_1. The additive probability error $2^{-\Omega(\lambda)}$ is caused by trivial subgroup components in T. Because we actually take T as a generator, our simulation will deviate from both or one of the games if there exists any trivial subgroup component in it. □

Proof of Lemma 4.

Proof. Given $(\mathcal{G}, g_1, g_4, X_1X_2X_3, T)$ where either $T = u \leftarrow \mathbb{G}_{p_1}$ or $T = ug_2^r \leftarrow \mathbb{G}_{p_1}\mathbb{G}_{p_2}$ for $g_2 \leftarrow \mathbb{G}_{p_2}^*$ and $r \leftarrow \mathbb{Z}_N$, algorithm $\mathcal{B}$ works as follows:

Initialization. Pick $\alpha \leftarrow \mathbb{Z}_N$ and select hash function H. Output

$$\text{MPK} = (g_1,\ g_1^{\alpha},\ e(g_1, T),\ H).$$

Observe that $e(g_1, T) = e(g_1, u)$ in both cases.

Answering O^{KeyGen}. On input (ID), sample $R_4 \leftarrow \mathbb{G}_{p_4}$ and return

$$T^{\frac{1}{\alpha+\text{ID}}} \cdot R_4.$$

Answering O_β^{Enc}. On input (ID^*), sample $s' \leftarrow \mathbb{Z}_N$ and compute

$$\text{CT}_1^* = (X_1X_2X_3)^{s'} \quad \text{and} \quad \text{KEY}_1^* = H(e((X_1X_2X_3)^{s'}, \text{SK}_{\text{ID}^*}))$$

where SK_{ID^*} is obtained via oracle O^{KeyGen}. $\mathcal{B}$ then picks $(\text{CT}_0^*, \text{KEY}_0^*) \leftarrow \mathbb{G}_{p_1} \times \{0,1\}^\lambda$ and returns $(\text{CT}_\beta^*, \text{KEY}_\beta^*)$.

Finalize. $\mathcal{B}$ returns 1 if $\beta = \beta'$ and returns 0 in the other case.

When $T = u$, the simulation is identical to Game_1; when $T = ug_2^r$, the simulation is identical to $\mathsf{Game}_{2.0}$ where $\alpha_1 = \alpha \bmod p_2$ and $r_1 = r \bmod p_2$. The additive probability error $2^{-\Omega(\lambda)}$ is caused by trivial subgroup components in $X_1X_2X_3$. □

Proof of Lemma 5.

Proof. Given $(\mathcal{G}, g_1, g_4, X_1X_2X_3, T)$ where either $T = g_2 \leftarrow \mathbb{G}_{p_2}$ or $T = g_2g_3 \leftarrow \mathbb{G}_{p_2}\mathbb{G}_{p_3}$, algorithm $\mathcal{B}$ works as follows:

Initialization. Pick $\alpha \leftarrow \mathbb{Z}_N$ and $u \leftarrow \mathbb{G}_{p_1}$. Select hash function H. Output

$$\text{MPK} = (g_1,\ g_1^{\alpha},\ e(g_1, u),\ H).$$

Sample $\alpha_1', \ldots, \alpha_{2^i}', r_1', \ldots, r_{2^i}' \leftarrow \mathbb{Z}_N$.

Answering O^{KeyGen}. On input (ID), sample $R_4 \leftarrow \mathbb{G}_{p_4}$ and return

$$u^{\frac{1}{\alpha+\text{ID}}} \cdot T^{\sum_{j=1}^{2^i} \frac{r_j'}{\alpha_j'+\text{ID}}} \cdot R_4.$$

Answering O_β^{Enc}. On input (ID^*), sample $s' \leftarrow \mathbb{Z}_N$ and compute

$$\text{CT}_1^* = (X_1X_2X_3)^{s'} \quad \text{and} \quad \text{KEY}_1^* = H(e((X_1X_2X_3)^{s'}, \text{SK}_{\text{ID}^*}))$$

where SK_{ID^*} is obtained via oracle O^{KeyGen}. $\mathcal{B}$ then picks $(\text{CT}_0^*, \text{KEY}_0^*) \leftarrow \mathbb{G}_{p_1} \times \{0,1\}^\lambda$ and returns $(\text{CT}_\beta^*, \text{KEY}_\beta^*)$.

Finalize. $\mathcal{B}$ returns 1 if $\beta = \beta'$ and returns 0 in the other case.

When $T = g_2$, the simulation is identical to $\mathsf{Game}_{2.i}$; when $T = g_2g_3$, the simulation is identical to $\mathsf{Game}_{2.i.1}$. We set

$$\alpha_j = \alpha_j' \bmod p_2,\ r_j = r_j' \bmod p_2, \quad \text{for all } j \in [2^i]$$

for both cases and set

$$\widehat{\alpha}_j = \alpha_j' \bmod p_3,\ \widehat{r}_j = r_j' \bmod p_3, \quad \text{for all } j \in [2^i]$$

in the case of $T = g_2g_3$. The additive probability error $2^{-\Omega(\lambda)}$ is caused by trivial subgroup components in $X_1X_2X_3$ and T. □

Proof of Lemma 6.

Proof. Given $(\mathcal{G}, g_1, g_4, X_1X_2X_3, Y_2Y_4, T)$ where either $T = g_2R_4 \leftarrow \mathbb{G}_{p_2}\mathbb{G}_{p_4}$ or $T = g_3R_4 \leftarrow \mathbb{G}_{p_3}\mathbb{G}_{p_4}$, algorithm $\mathcal{B}$ works as follows:

Initialization. Pick $\alpha \leftarrow \mathbb{Z}_N$ and $u \leftarrow \mathbb{G}_{p_1}$. Select hash function H. Output

$$\text{MPK} = (g_1,\ g_1^{\alpha},\ e(g_1, u),\ H).$$

Sample $\alpha'_1, \ldots, \alpha'_{2^i}, r'_1, \ldots, r'_{2^i}, \widehat{\alpha}_1, \ldots, \widehat{\alpha}_{2^i}, \widehat{r}_1, \ldots, \widehat{r}_{2^i} \leftarrow \mathbb{Z}_N$.

Answering O^{KeyGen}. On input (ID), sample $R'_4 \leftarrow \mathbb{G}_{p_4}$ and return

$$u^{\frac{1}{\alpha+\text{ID}}} \cdot (Y_2Y_4)^{\sum_{j=1}^{2^i} \frac{r'_j}{\alpha'_j+\text{ID}}} \cdot T^{\sum_{j=1}^{2^i} \frac{\widehat{r}_j}{\widehat{\alpha}_j+\text{ID}}} \cdot R'_4.$$

Answering O_β^{Enc}. On input (ID^*), sample $s' \leftarrow \mathbb{Z}_N$ and compute

$$\text{CT}_1^* = (X_1X_2X_3)^{s'} \quad \text{and} \quad \text{KEY}_1^* = H(e((X_1X_2X_3)^{s'}, \text{SK}_{\text{ID}^*}))$$

where SK_{ID^*} is obtained via oracle O^{KeyGen}. $\mathcal{B}$ then picks $(\text{CT}_0^*, \text{KEY}_0^*) \leftarrow \mathbb{G}_{p_1} \times \{0,1\}^\lambda$ and returns $(\text{CT}_\beta^*, \text{KEY}_\beta^*)$.

Finalize. $\mathcal{B}$ returns 1 if $\beta = \beta'$ and returns 0 in the other case.

Let $Y_2Y_4 = g_2^{y_2}g_4^{y_4}$, we implicitly set

$$\alpha_j = \alpha'_j \bmod p_2 \quad \text{and} \quad r_j = r'_j \cdot y_2 \bmod p_2 \quad \text{for all } j \in [2^i].$$

When $T = g_3R_4$, the simulation is identical to $\mathsf{Game}_{2.i.1}$; when $T = g_2R_4$, the simulation is identical to $\mathsf{Game}_{2.i.2}$. The additive probability error $2^{-\Omega(\lambda)}$ is caused by trivial subgroup components in $X_1X_2X_3$, Y_2Y_4 and T. □

4 Towards *Full* Adaptive Security

To prove the full adaptive security of our IBE scheme (in the single-instance setting), we still employ the game sequence described in the previous section. In fact, *nearly* all lemmas and results we have established still hold in the *full* adaptive security model where each target identity may have more than one challenge ciphertext. The only exception is that we can not prove the indistinguishability between Game_3 and Game_4 as before. (See Section 1.2 for an explanation.)

Following the work by Hofheinz *et al.* [HKS15], we find that we can prove the indistinguishability between them under a subgroup variant of DBDH assumption (see Assumption 5). This assumption is motivated by *Dual System Bilinear DDH assumption* from [HKS15] but is simpler.

Assumption 5 (DBDH in $\mathbb{G}_{p_2}$) *For any p.p.t. adversary $\mathcal{A}$ the following advantage function is negligible in λ.*

$$\mathsf{Adv}_{\mathcal{A}}^{\text{DBDH}}(\lambda) = |\Pr[\mathcal{A}(\mathcal{G}, D, T_0) = 1] - \Pr[\mathcal{A}(\mathcal{G}, D, T_1)]|,$$

where $\mathcal{G} \leftarrow \mathsf{GrpGen}(1^\lambda)$, $g_1 \leftarrow \mathbb{G}_{p_1}^*$, $g_2 \leftarrow \mathbb{G}_{p_2}^*$, $g_3 \leftarrow \mathbb{G}_{p_3}^*$, $g_4 \leftarrow \mathbb{G}_{p_4}^*$, $a, b, c, r \leftarrow \mathbb{Z}_N$,

$$D = (\mathcal{G}, g_1, g_3, g_4, g_2, g_2^a, g_2^b, g_2^c);$$
$$T_0 = e(g_2, g_2)^{abc} \quad \textit{and} \quad T_1 \leftarrow e(g_2, g_2)^r.$$

We can define two efficient algorithms to re-randomize DBDH problem instances as Hofheinz *et al.* [HKS15]. Given a DBDH instance, algorithm ReRand produces an entirely fresh instance while algorithm ReRand_a creates a fresh instance sharing b and c with its input. Their formal definitions are given below.

– $\mathsf{ReRand}_a(g_2, g_2^a, g_2^b, g_2^c, T) \to (g_2^{a'}, T')$ where $a' \leftarrow \mathbb{Z}_N$ and

$$T' = \begin{cases} e(g_2, g_2)^{a'bc} & \text{when } T = e(g_2, g_2)^{abc} \\ e(g_2, g_2)^{r'} \text{ for } r' \leftarrow \mathbb{Z}_N & \text{when } T = e(g_2, g_2)^{r} \end{cases}$$

– $\mathsf{ReRand}(g_2, g_2^a, g_2^b, g_2^c, T) \to (g_2^{a'}, g_2^{b'}, g_2^{c'}, T')$ where $a', b', c' \leftarrow \mathbb{Z}_N$ and

$$T' = \begin{cases} e(g_2, g_2)^{a'b'c'} & \text{when } T = e(g_2, g_2)^{abc} \\ e(g_2, g_2)^{r'} \text{ for } r' \leftarrow \mathbb{Z}_N & \text{when } T = e(g_2, g_2)^{r} \end{cases}$$

We now prove that Game_3 and Game_4 are *computationally* indistinguishable in the *full* adaptive security model. This will immediately derive the full adaptive security of our IBE scheme in the single-instance setting.

Lemma 7 (from Game_3 to Game_4). *For any p.p.t. adversary $\mathcal{A}$ sending at most q_σ queries to O^{KeyGen} and O_β^{Enc}, there exists $\mathcal{B}$ with $T(\mathcal{B}) \approx T(\mathcal{A}) + q_\sigma \cdot \mathsf{poly}(\lambda)$ and*

$$|\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_3}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_4}(\lambda)| \leqslant \mathsf{Adv}_{\mathcal{B}}^{\mathrm{DBDH}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Proof. Given $(\mathcal{G}, g_1, g_3, g_4, g_2, g_2^a, g_2^b, g_2^c, T)$ where either $T = e(g_2, g_2)^{abc}$ or $T = e(g_2, g_2)^r$ for some $r \leftarrow \mathbb{Z}_N$, algorithm $\mathcal{B}$ works as follows:

Initialization. Pick $\alpha \leftarrow \mathbb{Z}_N$ and $u \leftarrow \mathbb{G}_{p_1}$. Select hash function H. Output

$$\mathrm{MPK} = (g_1,\ g_1^\alpha,\ e(g_1, u),\ H).$$

We maintain random function RF in an on-the-fly way.

Answering O^{KeyGen}. On input (ID), return

$$u^{\frac{1}{\alpha + \mathrm{ID}}} \cdot g_2^{\mathsf{RF}(\mathrm{ID})} \cdot R_4$$

where $R_4 \leftarrow \mathbb{G}_{p_4}$ and RF is a truly random function.

Answering O_β^{Enc}. $\mathcal{B}$ maintains a list $\mathcal{L}$. On input (ID^*), sample $s' \leftarrow \mathbb{Z}_N$. If one can find a entry $(\mathrm{ID}^*, g_2^{a'}, g_2^{b'}, g_2^{c'}, T') \in \mathcal{L}$, get

$$(g_2^{a^*}, T^*) \leftarrow \mathsf{ReRand}_a(g_2^{a'}, g_2^{b'}, g_2^{c'}, T');$$

otherwise get

$$(g_2^{a^*}, g_2^{b^*}, g_2^{c^*}, T^*) \leftarrow \mathsf{ReRand}(g_2^a, g_2^b, g_2^c, T)$$

and update the list as $\mathcal{L} = \mathcal{L} \cup \{(\mathrm{ID}^*, g_2^{a^*}, g_2^{b^*}, g_2^{c^*}, T^*)\}$. $\mathcal{B}$ then computes

$$\mathrm{CT}_1^* = (g_1 g_3)^{s'} \cdot g_2^{a^*} \quad \text{and} \quad \mathrm{KEY}_1^* = H(e(g_1^{s'}, u^{\frac{1}{\alpha + \mathrm{ID}^*}}) \cdot T^*).$$

Finally $\mathcal{B}$ picks $(\mathrm{CT}_0^*, \mathrm{KEY}_0^*) \leftarrow \mathbb{G}_{p_1} \times \{0,1\}^\lambda$ and returns $(\mathrm{CT}_\beta^*, \mathrm{KEY}_\beta^*)$.

Finalize. $\mathcal{B}$ returns 1 if $\beta = \beta'$ and returns 0 in the other case.

We implicitly define RF as

$$\mathsf{RF}(\mathrm{ID}^*) = b^* c^* \quad \text{for all } (\mathrm{ID}^*, g_2^{a^*}, g_2^{b^*}, g_2^{c^*}, T^*) \in \mathcal{L} \text{ (or } \mathrm{ID}^* \in Q_C).$$

Since it is ensured that $\mathrm{ID}^* \notin Q_K$ for all $(\mathrm{ID}^*, \star, \star, \star, \star) \in \mathcal{L}$ (or $\mathrm{ID}^* \in Q_C$), our simulation of RF is consistent. When $T = e(g_2, g_2)^{abc}$, the simulation is identical to Game_3 where

$$T^* = e(g_2^{a^*}, g_2^{\mathsf{RF}(\mathrm{ID}^*)});$$

when $T = e(g_2, g_2)^r$ for some $r \leftarrow \mathbb{Z}_N$, the simulation is identical to Game_4 since all inputs of H have min-entropy $\Theta(\lambda)$ and thus distributions of all KEY_1^* are statistically close to the uniform distribution over $\{0,1\}^\lambda$. □

5 Towards *Multi-instance* Setting

Having obtained full adaptive security of our IBE scheme in the basic single-instance setting, we now extend the result to the *multi-instance setting* [HKS15]. Typically, all instances in question will share some parameters. Formally, we define two additional algorithms following [HKS15]:

- $\mathsf{Param}(1^\lambda) \to \mathrm{GP}$. The *parameter generation algorithm* Param takes as input the security parameter 1^λ and outputs global parameter GP.
- $\mathsf{Setup}_\mathsf{m}(\mathrm{GP}) \to (\mathrm{MPK}, \mathrm{MSK})$. The *setup algorithm* $\mathsf{Setup}_\mathsf{m}$ takes as input the global parameter GP and outputs master public/secret key pair $(\mathrm{MPK}, \mathrm{MSK})$.

Each instance is established by running algorithm $\mathsf{Setup}_\mathsf{m}$ with the global parameter GP (shared among all instances) and a fresh random coin. For simplicity, we assume that all instances have common ciphertext space $\mathcal{C}$ and key space $\mathcal{K}$. With master public/secret key pair $(\mathrm{MPK}, \mathrm{MSK})$ generated by algorithm $\mathsf{Setup}_\mathsf{m}$, one can invoke algorithms KeyGen, Enc, Dec as in the single-instance setting. Therefore the correctness can be defined in a natural way.

The full adaptive security and anonymity in the multi-instance setting can be formulated by defining the advantage function as

$$\mathsf{Adv}^{\mathrm{mIBE}}_{\mathcal{A}}(\lambda) = \left| \Pr\left[\beta = \beta' \;\middle|\; \begin{array}{c} \mathrm{GP} \leftarrow \mathsf{Param}(1^\lambda),\ \beta \leftarrow \{0,1\} \\ (\mathrm{MPK}^{(\iota)}, \mathrm{MSK}^{(\iota)}) \leftarrow \mathsf{Setup}_\mathsf{m}(\mathrm{GP}), \quad \forall \iota \in [\tau] \\ \beta' \leftarrow \mathcal{A}^{\mathsf{O}^{\mathsf{KeyGen}}(\cdot,\cdot), \mathsf{O}^{\mathsf{Enc}}_\beta(\cdot,\cdot)}(1^\lambda, \mathrm{MPK}^{(1)}, \ldots, \mathrm{MPK}^{(\tau)}) \end{array} \right] - \frac{1}{2} \right|$$

where τ is the number of instances and oracles work as follows

- $\mathsf{O}^{\mathsf{KeyGen}}$: On input (ι, ID), the oracle returns $\mathsf{KeyGen}(\mathrm{MPK}^{(\iota)}, \mathrm{MSK}^{(\iota)}, \mathrm{ID})$ and sets $\mathrm{Q_K} = \mathrm{Q_K} \cup \{(\iota, \mathrm{ID})\}$.
- $\mathsf{O}^{\mathsf{Enc}}_\beta$: On input (ι^*, ID^*), the oracle samples $(\mathrm{CT}^*_1, \mathrm{KEY}^*_1) \leftarrow \mathsf{Enc}(\mathrm{MPK}^{(\iota^*)}, \mathrm{ID}^*)$, $(\mathrm{CT}^*_0, \mathrm{KEY}^*_0) \leftarrow \mathcal{C} \times \mathcal{K}$ and returns $(\mathrm{CT}^*_\beta, \mathrm{KEY}^*_\beta)$. Set $\mathrm{Q_C} = \mathrm{Q_C} \cup \{(\iota^*, \mathrm{ID}^*)\}$.

5.1 Construction

We describe a multi-instance variant of our basic IBE scheme (in Section 3.1) as follows.

- $\mathsf{Param}(1^\lambda)$. Run $\mathcal{G} = (N, G, G_T, e) \leftarrow \mathsf{GrpGen}(1^\lambda)$. Sample

$$g_1 \leftarrow G^*_{p_1}, \quad g_4 \leftarrow G^*_{p_4}.$$

  Pick $H : G_T \to \{0,1\}^\lambda$ from a pairwise independent hash family. Output

$$\mathrm{GP} = (\mathcal{G},\ g_1,\ g_4,\ H).$$

- $\mathsf{Setup}_\mathsf{m}(\mathrm{GP})$. Sample $\alpha \leftarrow \mathbb{Z}_N$ and $u \leftarrow G_{p_1}$. Output

$$\mathrm{MPK} = (g_1,\ g_1^\alpha,\ e(g_1, u),\ H) \quad \text{and} \quad \mathrm{MSK} = (\alpha, u, g_4).$$

The remaining algorithms KeyGen, Enc, Dec are defined as in Section 3.1.

5.2 Security

We prove the following theorem.

Theorem 2. *For any p.p.t. adversary $\mathcal{A}$ sending at most $\hat{q}_\sigma$ queries to $\mathsf{O}^{\mathsf{KeyGen}}$ and $\mathsf{O}^{\mathsf{Enc}}_\beta$ for each of τ instances, there exist $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4$ such that*

$$\mathsf{Adv}^{\mathrm{mIBE}}_{\mathcal{A}}(\lambda) \leqslant \frac{5}{2} \cdot \mathsf{Adv}^{\mathrm{SD1}}_{\mathcal{B}_1}(\lambda) + 2 \cdot \mathsf{Adv}^{\mathrm{SD2}}_{\mathcal{B}_2}(\lambda) + 2 \cdot \lceil \log \hat{q}_\sigma \rceil \cdot \mathsf{Adv}^{\mathrm{SD3}}_{\mathcal{B}_3}(\lambda) + \\ + \left(2 \cdot \lceil \log \hat{q}_\sigma \rceil + \frac{1}{2}\right) \cdot \mathsf{Adv}^{\mathrm{SD4}}_{\mathcal{B}_4}(\lambda) + 2^{-\Omega(\lambda)}$$

and $\max\{\mathsf{T}(\mathcal{B}_1), \mathsf{T}(\mathcal{B}_2), \mathsf{T}(\mathcal{B}_3), \mathsf{T}(\mathcal{B}_4)\} \approx \mathsf{T}(\mathcal{A}) + \tau^2 \cdot q_\sigma^2 \cdot \mathsf{poly}(\lambda)$.

One may find that the above theorem is almost the same as Theorem 1. As a matter of fact, it can be proved in a similar way. As we have discussed in Section 1.2, our main idea is to build an *independent* random function for *each* instance in a *concurrent* manner. The remaining of this subsection is devoted to showing how to upgrade the proof of Theorem 1 (c.f. Section 3.2 for game sequence and Section 3.3 for proof details) to prove Theorem 2.

Game Sequence. It's quite straightforward to extend $\mathsf{Game}_{\mathsf{real}}$, Game_0, Game'_0, Game''_0, Game_1 and Game_4 to the multi-instance setting. The remaining $\mathsf{Game}_{2.i}$, $\mathsf{Game}_{2.i.1}$, $\mathsf{Game}_{2.i.2}$, for $0 \leqslant i \leqslant \lceil \log \hat{q}_\sigma \rceil$, and Game_3 can be described as follows: Let $\mathcal{G} = (N, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathsf{GrpGen}(1^\lambda)$. In all these games, master public keys given to adversary $\mathcal{A}$ are

$$\mathrm{MPK}^{(1)} = (g_1,\ g_1^{\alpha^{(1)}},\ e(g_1, u^{(1)}),\ \mathsf{H}), \quad \ldots, \quad \mathrm{MPK}^{(\tau)} = (g_1,\ g_1^{\alpha^{(\tau)}},\ e(g_1, u^{(\tau)}),\ \mathsf{H})$$

where $g_1 \leftarrow \mathbb{G}_{p_1}^*$, $\alpha^{(1)}, \ldots, \alpha^{(\tau)} \leftarrow \mathbb{Z}_N$, $u^{(1)}, \ldots, u^{(\tau)} \leftarrow \mathbb{G}_{p_1}$ and H is picked from a family of pairwise-independent hash family; oracle $\mathsf{O}^{\mathsf{Enc}}_\beta$ works as follows:

- On input (ι^*, ID^*), sample $\mathrm{CT}^*_1 \leftarrow \mathbb{G}_{p_1}\mathbb{G}_{p_2}\mathbb{G}_{p_3}$ and compute
$$\mathrm{KEY}^*_1 = \mathsf{H}(e(\mathrm{CT}^*_1, \mathrm{SK}^{(\iota^*)}_{\mathrm{ID}^*}))$$
where $\mathrm{SK}^{(\iota^*)}_{\mathrm{ID}^*}$ is obtained via a $\mathsf{O}^{\mathsf{KeyGen}}$ query (ι^*, ID^*). Sample $(\mathrm{CT}^*_0, \mathrm{KEY}^*_0) \leftarrow \mathbb{G}_{p_1} \times \{0,1\}^\lambda$ and return $(\mathrm{CT}^*_\beta, \mathrm{KEY}^*_\beta)$.

However, on input (ι, ID), oracle $\mathsf{O}^{\mathsf{KeyGen}}$ behaves differently in those games:

- In $\mathsf{Game}_{2.i}$, it returns
$$(u^{(\iota)})^{\frac{1}{\alpha^{(\iota)}+\mathrm{ID}}} \cdot \boxed{g_2^{\sum_{j=1}^{2^i} \frac{r_j^{(\iota)}}{\alpha_j^{(\iota)}+\mathrm{ID}}}} \cdot R_4$$
where $g_2 \leftarrow \mathbb{G}_{p_2}^*$ and $\alpha_j^{(1)}, r_j^{(1)}, \ldots, \alpha_j^{(\tau)}, r_j^{(\tau)} \leftarrow \mathbb{Z}_N$ for all $j \in [2^i]$.
- In $\mathsf{Game}_{2.i.1}$, it returns
$$(u^{(\iota)})^{\frac{1}{\alpha^{(\iota)}+\mathrm{ID}}} \cdot g_2^{\sum_{j=1}^{2^i} \frac{r_j^{(\iota)}}{\alpha_j^{(\iota)}+\mathrm{ID}}} \cdot \boxed{g_3^{\sum_{j=1}^{2^i} \frac{\hat{r}_j^{(\iota)}}{\hat{\alpha}_j^{(\iota)}+\mathrm{ID}}}} \cdot R_4,$$
where $g_3 \leftarrow \mathbb{G}_{p_3}^*$ and $\alpha_j^{(1)}, r_j^{(1)}, \hat{\alpha}_j^{(1)}, \hat{r}_j^{(1)}, \ldots, \alpha_j^{(\tau)}, r_j^{(\tau)}, \hat{\alpha}_j^{(\tau)}, \hat{r}_j^{(\tau)} \leftarrow \mathbb{Z}_N$ for all $j \in [2^i]$.
- In $\mathsf{Game}_{2.i.2}$, it returns
$$(u^{(\iota)})^{\frac{1}{\alpha^{(\iota)}+\mathrm{ID}}} \cdot g_2^{\sum_{j=1}^{2^i} \frac{r_j^{(\iota)}}{\alpha_j^{(\iota)}+\mathrm{ID}} + \boxed{\sum_{j=1}^{2^i} \frac{\hat{r}_j^{(\iota)}}{\hat{\alpha}_j^{(\iota)}+\mathrm{ID}}}} \cdot R_4,$$
where $g_2 \leftarrow \mathbb{G}_{p_2}^*$ and $\alpha_j^{(1)}, r_j^{(1)}, \hat{\alpha}_j^{(1)}, \hat{r}_j^{(1)}, \ldots, \alpha_j^{(\tau)}, r_j^{(\tau)}, \hat{\alpha}_j^{(\tau)}, \hat{r}_j^{(\tau)} \leftarrow \mathbb{Z}_N$ for all $j \in [2^i]$.
- In Game_3, it returns
$$(u^{(\iota)})^{\frac{1}{\alpha^{(\iota)}+\mathrm{ID}}} \cdot g_2^{\mathsf{RF}^{(\iota)}(\mathrm{ID})} \cdot R_4$$
where $g_2 \leftarrow \mathbb{G}_{p_2}^*$ and $\mathsf{RF}^{(1)}, \ldots, \mathsf{RF}^{(\tau)}$ are τ *independent* random functions.

Lemmas and Proofs. Most lemmas and proofs (including arguments) in Section 3.2, Section 3.3 and Section 4 can be extended directly to cope with multiple instances. In particular, in order to prove $\mathsf{Game}_{2.i} \approx \mathsf{Game}_{2.i.1}$, $\mathsf{Game}_{2.i.1} \approx \mathsf{Game}_{2.i.2}$, and $\mathsf{Game}_3 \approx \mathsf{Game}_4$ (where "$\mathsf{Game}_{xxx} \approx \mathsf{Game}_{yyy}$" means two games are computationally indistinguishable) in the multi-instance setting, one can just invoke simulators described in the proofs of Lemma 5, Lemma 6, and Lemma 7 for *each* instance using *independent* random coins. It remains to give the following lemma showing $\mathsf{Game}_1 \approx \mathsf{Game}_{2.0}$ with proof.

Lemma 8 (from Game_1 to $\mathsf{Game}_{2.0}$, multi-instance case). *For any p.p.t. adversary $\mathcal{A}$ sending at most $\hat{q}_\sigma$ queries to $\mathsf{O}^{\mathsf{KeyGen}}$ and $\mathsf{O}^{\mathsf{Enc}}_\beta$ for each of τ instances, there exists $\mathcal{B}$ with $\mathsf{T}(\mathcal{B}) \approx \mathsf{T}(\mathcal{A}) + \tau \cdot \hat{q}_\sigma \cdot \mathsf{poly}(\lambda)$ and*

$$|\mathsf{Adv}^{\mathsf{Game}_{2.0}}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{Game}_1}_{\mathcal{A}}(\lambda)| \leqslant \mathsf{Adv}^{\mathsf{SD2}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}.$$

Proof. Given $(\mathcal{G}, g_1, g_4, X_1X_2X_3, T)$ where either $T = g_1^\mu \leftarrow \mathbb{G}_{p_1}$ or $T = g_1^\mu g_2^r \leftarrow \mathbb{G}_{p_1}\mathbb{G}_{p_2}$ for $g_2 \leftarrow \mathbb{G}^*_{p_2}$ and $\mu, r \leftarrow \mathbb{Z}_N$, algorithm $\mathcal{B}$ works as follows:

Initialization. Pick $\alpha^{(1)}, \ldots, \alpha^{(\tau)}, \mu^{(1)}, \ldots, \mu^{(\tau)} \leftarrow \mathbb{Z}_N$ and select hash function H. Compute

$$T^{(1)} = T^{\mu^{(1)}}, \ \ldots \ , \ T^{(\tau)} = T^{\mu^{(\tau)}}$$

and output

$$\mathrm{MPK}^{(1)} = (g_1,\ g_1^{\alpha^{(1)}},\ e(g_1, T^{(1)}),\ H),\ \ldots,\ \mathrm{MPK}^{(\tau)} = (g_1,\ g_1^{\alpha^{(\tau)}},\ e(g_1, T^{(\tau)}),\ H).$$

Here we implicitly set

$$u^{(1)} = g_1^{\mu\mu^{(1)}}, \ \ldots \ , \ u^{(\tau)} = g_1^{\mu\mu^{(\tau)}}.$$

Answering $\mathsf{O}^{\mathsf{KeyGen}}$. On input (ι, ID), sample $R_4 \leftarrow \mathbb{G}_{p_4}$ and return

$$(T^{(\iota)})^{\frac{1}{\alpha^{(\iota)}+\mathrm{ID}}} \cdot R_4.$$

Answering $\mathsf{O}^{\mathsf{Enc}}_\beta$. On input (ι^*, ID^*), sample $s' \leftarrow \mathbb{Z}_N$ and compute

$$\mathrm{CT}^*_1 = (X_1X_2X_3)^{s'} \quad \text{and} \quad \mathrm{KEY}^*_1 = H(e((X_1X_2X_3)^{s'}, \mathrm{SK}_{\mathrm{ID}^*}))$$

where $\mathrm{SK}_{\mathrm{ID}^*}$ is obtained via a $\mathsf{O}^{\mathsf{KeyGen}}$ query. $\mathcal{B}$ then picks $(\mathrm{CT}^*_0, \mathrm{KEY}^*_0) \leftarrow \mathbb{G}_{p_1} \times \{0,1\}^\lambda$ and returns $(\mathrm{CT}^*_\beta, \mathrm{KEY}^*_\beta)$.

Finalize. $\mathcal{B}$ returns 1 if $\beta = \beta'$ and returns 0 in the other case.

When $T = g_1^\mu$, the simulation is identical to Game_1; when $T = g_1^\mu g_2^r$, the simulation is identical to $\mathsf{Game}_{2.0}$ where we implicitly set

$$\begin{matrix} \alpha_1^{(1)} = \alpha^{(1)} \bmod p_2 \\ r_1^{(1)} = r\mu^{(1)} \bmod p_2 \end{matrix}, \ \cdots \ , \ \begin{matrix} \alpha_1^{(\tau)} = \alpha^{(\tau)} \bmod p_2 \\ r_1^{(\tau)} = r\mu^{(\tau)} \bmod p_2 \end{matrix}.$$

This proves the lemma. □

Acknowledgement. We thank Benoît Libert, Somindu Ramanna and Kai Zhang for their advices. We also greatly thank all anonymous reviewers of PKC 2017. Their constructive comments motivated us to extend our basic result to the multi-instance setting and helped us to clarify some technical subtlety.

References

AC16. Shashank Agrawal and Melissa Chase. A study of pair encodings: Predicate encryption in prime order groups. In *TCC 2016-A*, 2016. 7

AHY15. Nuttapong Attrapadung, Goichiro Hanaoka, and Shota Yamada. A framework for identity-based encryption with almost tight security. In *ASIACRYPT 2015*, 2015. 2, 3

Att14. Nuttapong Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In *EUROCRYPT 2014*. Springer, 2014. 7

Att16. Nuttapong Attrapadung. Dual system encryption framework in prime-order groups. *ASIACRYPT 2016*, 2016. 7

BB04a. Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *EUROCRYPT 2004*, 2004. 2

BB04b. Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In *CRYPTO 2004*, 2004. 2

BF01. Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In *CRYPTO 2001*, 2001. 2, 3

BGN05. Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-dnf formulas on ciphertexts. In *TCC 2005*, 2005. 3, 8

BKP14. Olivier Blazy, Eike Kiltz, and Jiaxin Pan. (Hierarchical) Identity-based encryption from affine message authentication. In *CRYPTO 2014*, 2014. 2

BL16. Xavier Boyen and Qinyi Li. Towards tightly secure lattice short signature and Id-based encryption. In *ASIACRYPT 2016*, 2016. 7

BR93. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS '93*, 1993. 2

BWY11. Mihir Bellare, Brent Waters, and Scott Yilek. Identity-based encryption secure against selective opening attack. In *TCC 2011*, 2011. 3, 8

CGW15. Jie Chen, Romain Gay, and Hoeteck Wee. Improved dual system abe in prime-order groups via predicate encodings. In *EUROCRYPT 2015*, 2015. 7

CLR16. Jie Chen, Benoît Libert, and Somindu C. Ramanna. Non-zero inner product encryption with short ciphertexts and private keys. In *SCN 2016*, 2016. 7

CM14. Melissa Chase and Sarah Meiklejohn. Déjà Q: Using dual systems to revisit q-type assumptions. In *EUROCRYPT 2014*, 2014. 4, 5, 7

CMM16. Melissa Chase, Mary Maller, and Sarah Meiklejohn. Déjà Q all over again: Tighter and broader reductions of q-type assumptions. In *ASIACRYPT 2016*, 2016. 7

CW13. Jie Chen and Hoeteck Wee. Fully,(almost) tightly secure ibe and dual system groups. In *CRYPTO 2013*, 2013. 2, 3, 5, 7

GCD+16. Junqing Gong, Jie Chen, Xiaolei Dong, Zhenfu Cao, and Shaohua Tang. Extended nested dual system groups, revisited. In *PKC 2016*, 2016. 2

GDCC16. Junqing Gong, Xiaolei Dong, Jie Chen, and Zhenfu Cao. Efficient IBE with tight reduction to standard assumption in the multi-challenge setting. In *ASIACRYPT 2016*, 2016. 2

Gen06. Craig Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT 2006*, 2006. 2

HKS15. Dennis Hofheinz, Jessica Koch, and Christoph Striecks. Identity-based encryption with (almost) tight security in the multi-instance, multi-ciphertext setting. In *PKC 2015*. Springer, 2015. 2, 3, 6, 15, 17

Hof16a. Dennis Hofheinz. Adaptive partitioning. *IACR Cryptology ePrint Archive*, 2016, 2016. 7

Hof16b. Dennis Hofheinz. Algebraic partitioning: Fully compact and (almost) tightly secure cryptography. In *TCC 2016*, 2016. 7

LOS+10. Allison Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *CRYPTO 2010*. S, 2010. 7

LRY16. Benoît Libert, Somindu C. Ramanna, and Moti Yung. Functional commitment schemes: From polynomial commitments to pairing-based accumulators from simple assumptions. In *ICALP 2016*, 2016. 7

NR04. Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. *J. ACM*, 51(2):231–262, 2004. 2, 3

OT10. Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In *CRYPTO 2010*, 2010. 7

Sha84. Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO 84*, 1984. 2

Wat05. Brent Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT 2005*, 2005. 2

Wat09. Brent Waters. Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In *CRYPTO 2009*, 2009. 2, 4

Wee14. Hoeteck Wee. Dual system encryption via predicate encodings. In *TCC 2014*, 2014. 7

Wee16. Hoeteck Wee. Déjà Q: Encore! un petit ibe. In *TCC 2016*, 2016. 3, 4, 5, 6, 7, 8, 9