Towards Forward Security Properties for PEKS and IBE

Qiang Tang

SnT, University of Luxembourg
6, rue Richard Coudenhove-Kalergi, L-1359 Luxembourg
qiang.tang@uni.lu

Abstract. In cryptography, forward secrecy is a well-known property for key agreement protocols. It ensures that a session key will remain private even if one of the long-term secret keys is compromised in the future. In this paper, we investigate some forward security properties for Public-key Encryption with Keyword Search (PEKS) schemes, which allow a client to store encrypted data and delegate search operations to a server. The proposed properties guarantee that the client's privacy is protected to the maximum extent even if his private key is compromised in the future. Motivated by the generic transformation from anonymous Identity-Based Encryption (IBE) to PEKS, we correspondingly propose some forward security properties for IBE, in which case we assume the attacker learns the master secret key. We then study several existing PEKS and IBE schemes, including a PEKS scheme by Nishioka, an IBE scheme by Boneh, Raghunathan and Segev, and an IBE scheme by Arriaga, Tang and Ryan. Our analysis indicates that the proposed forward security properties can be achieved by some of these schemes if the attacker is RO-non-adaptive (the attacker does not define its distributions based on the random oracle). Finally, we propose the concept of correlated-input indistinguishable hash function and show how to extend the Boyen-Waters anonymous IBE scheme to achieve the forward security properties against adaptive attackers.

1 Introduction

In the seminal work [8], Boneh et al. proposed the concept of Public-key Encryption with Keyword Search (PEKS) and formulated it as a cryptographic primitive with four algorithms (KeyGen, $\mathsf{Encrypt}$, $\mathsf{TrapGen}$, Test). PEKS is a two-party (i.e. client-server) primitive aiming at protecting a client's, say Alice's, privacy in the following *encrypted email routing scenario*.

1. Alice runs the KeyGen algorithm to generate a key pair (PK, SK) and publishes PK.
2. When any user, say Bob, sends an email to Alice, he can generate a tag $\mathsf{Encrypt}(x, PK)$ for a keyword x and attach it to the email (the email should be encrypted independently and the detail is omitted here). In the view of the email server, it has a list of emails indexed by $\mathsf{Encrypt}(x_1, PK)$, $\mathsf{Encrypt}(x_2, PK)$, $\cdots$ respectively.

3. If Alice wants to retrieve those emails indexed with a keyword y, she sends a trapdoor $\mathsf{TrapGen}(y, SK)$ to the email server, which can then run an algorithm Test on the input
   $(\mathsf{TrapGen}(y, SK), \mathsf{Encrypt}(x_i, PK))$ for every $i \geq 1$ to figure out whether $y = x_i$.

1.1 Problem Statement

With a PEKS scheme implemented, the server receives a list of tags from message senders and a list of trapdoors from the client. As a result of the desired search functionality, the server can try to match any tag with any trapdoor. Therefore, the server can categorize the possessed tags and trapdoors into three scenarios.

– Scenario 1: the tags, which do not match any trapdoor. The seminal work [8] and all follow-ups have considered the privacy of keywords in this scenario. It is worth noting that most of these papers only consider this property.

– Scenario 2: the tags and trapdoors, which match at least one trapdoor or tag. In [9] and its full version [10], Boneh, Raghunathan and Segev defined (enhanced) function privacy properties for the keywords in this scenario [1].

– Scenario 3: the trapdoors, which do not match any tag. In [4], Arriaga, Tang, and Ryan defined search pattern privacy properties which captures the privacy of keywords in this scenario.

It is clear that the above scenarios are mutually independent and their security properties will not be comparable (as already indicated in [4]). *As such, a PEKS scheme should provide maximal protection for the keywords in all three scenarios.* Unfortunately, the security properties defined in [4, 9] are rather weak and do not capture realistic threats.

– In the enhanced function privacy definition from [9], there is a min-entropy restriction on the distribution of keywords. It basically requires that if the keyword in a trapdoor $\mathsf{TrapGen}(y_i, SK)$ is different from those in $\mathsf{TrapGen}(y_1, SK)$, $\cdots$, $\mathsf{TrapGen}(y_{i-1}, SK)$, then it should be infeasible to guess y_i given $y_1, \cdots, y_{i-1}$. This restriction seems artificial and unrealistic. Taking the encrypted email routing scenario as an example, at a certain time period, the client may submit queries about a certain topic (e.g. work, family, or friends) and the keywords in the trapdoors might be highly correlated. This implies that, if some keywords are disclosed, it might be easy for the attacker to infer others.

– In the search pattern privacy property definition from [4], the distribution of keywords in the trapdoors is assumed to be uniform. This restriction is clearly unrealistic. Taking the encrypted email routing scenario as an example, it reasonable to expect the client to submit more queries with

[1] It is worth noting that the property for PEKS is not explicitly defined in [9, 10], but it is implied by their discussions (in fact, they use it to motivate the property definitions for IBE).

high-priority keywords such as ”urgent” than low-priority ones such as ”ordinary”, which implies that the keywords are not uniformly distributed.

Furthermore, it is possible that the client’s private key may get compromised at some point. If this happens, the client may still want the privacy of keywords in the tags and trapdoors to be preserved. This is similar to the forward secrecy requirement in key agreement protocols [19, 22]. However, no literature work has touched upon this property for PEKS.

1.2 Our Contribution

In this paper, we first introduce two new forward security properties for PEKS. One is forward-secure function privacy, which aims at protecting the privacy of keywords in Scenario 2. The other is forward-secure trapdoor unlinkability, which aims at protecting the privacy of keywords in Scenario 3. These two properties are much stronger than those from [9] and [4], in the sense that we not only allow the attacker to compromise the long-term secret key but also give it more flexibility to define the keyword distribution in the attack games. We analyse a PEKS scheme by Nishioka [28] and show that it only achieves our properties against RO-non-adaptive attackers (which can not choose the keyword distributions based on the random oracle).

We then introduce two new forward security properties for IBE, namely msk-forward-secure function privacy and msk-forward-secure key unlinkability, and they are augmented variants of those from [9] and [4] respectively. Naturally, the new properties directly lead to those forward security properties for PEKS as a result of the generic transformation proposed in [1]. We analyse the $\mathcal{IBE}_{\mathsf{DLIN2}}$ scheme by Boneh et al. [10], and show that it does not achieve the msk-forward-secure function privacy property. We also analyse an IBE scheme by Arriaga et al. [4], and show that it achieves our properties against RO-non-adaptive attackers.

Finally, we introduce the concept of correlated-input indistinguishable hash function, which can be regarded as an enhanced variant of the correlated-input secure hash functions proposed by Goyal, O’Neil, and Rao [21]. By preprocessing the identities with such a hash function, an IBE scheme automatically achieves the msk-forward-secure function privacy property against adaptive attackers. In contrast to the “extract-augment-combine” approach from [9], there is no need to tweak the encryption and decryption algorithms of the underlying IBE scheme. We then take Boyen-Waters anonymous IBE scheme [13] as an example, and extend it with composite order bilinear groups to achieve msk-forward-secure key unlinkability.

1.3 Organization

The rest of this paper is organized as follows. In Section 2, we present preliminaries on notation and hardness assumptions. In Section 3, we present an enhanced security model for PEKS with a focus on the forward security properties, and analyse the Nishioka scheme. In Section 4, we propose some forward security properties for IBE. In Section 5, we analyse an IBE scheme by Boneh et al. and an IBE scheme by Arriaga et al.. In Section 6, we introduce the concept of correlated-input indistinguishable hash function and extend the Boyen-Waters scheme. In Section 7, we review some related work. In Section 8, we conclude the paper.

2 Preliminary

2.1 Notation

- $x\|y$ means the concatenation of x and y, P.P.T. stands for probabilistic polynomial time.
- $x \xleftarrow{\$} \mathcal{A}^{O_1,O_2,\cdots}(m_1, m_2, \cdots)$ means that x is the output of the algorithm $\mathcal{A}$ which runs with the input $m_1, m_2, \cdots$ and has access to oracles $O_1, O_2, \cdots$.
- When X is a set, $x \xleftarrow{\$} X$ means that x is chosen from X uniformly at random, and $|X|$ means the size of X. When $\mathbb{D}$ is a distribution on the set X, $x \xleftarrow{\mathbb{D}} X$ means that x is a value sampled from X according to $\mathbb{D}$.
- We use bold letter, such as $\boldsymbol{X}$, to denote a vector or matrix. Given a vector $\boldsymbol{X}$, we use $\boldsymbol{X}^{(i)}$ to denote the i-th element in the vector. When g is a group element, we use $g^{\boldsymbol{X}}$ to denote a new vector or matrix, whose elements are exponentiations of the corresponding elements in $\boldsymbol{X}$. For two vectors (or matrices) $\boldsymbol{Y}$ and $\boldsymbol{Z}$ whose elements are from a group, we use $\boldsymbol{Y} \otimes \boldsymbol{Z}$ to denote the new vector (or matrix) after pairwise group operations.
- A function $P(\lambda) : \mathbb{Z} \to \mathbb{R}$ is said to be negligible with respect to λ if, for every polynomial $f(\lambda)$, there exists an integer N_f such that $P(\lambda) < \frac{1}{f(\lambda)}$ for all $\lambda \geq N_f$. When $P(\lambda)$ is negligible, then we say $1 - P(\lambda)$ is overwhelming.
- A random variable V has min-entropy λ, denoted as $H_\infty(V) = \lambda$, if $\max_v \Pr[V = v] = 2^{-\lambda}$, or equivalently $\lambda = -\log \max_v \Pr[V = v]$. If V has min-entropy at least λ, then V is a λ source. Given two random variables V and W, the conditional min-entropy of V with respect to W is defined to be $\min_w H_\infty(V|W = w)$, or equivalently $-\log \max_{v,w} \Pr[V = v|W = w]$.

2.2 Pairing over Composite-order Groups

A composite-order bilinear group generator is an algorithm $\mathcal{G}_C{}^{(pq)}$ that takes as input a security parameter λ and outputs a description $\Gamma = (p, q, \mathbb{G}, \mathbb{G}_T, \hat{e}, g_p, g_q)$ where:

- $\mathbb{G}$ and $\mathbb{G}_T$ are groups of order $n = pq$, where p and q are primes, with efficiently computable group laws.
- g_p is a randomly-chosen generator of the subgroup $\mathbb{G}_p$ of order p, and g_q is a randomly-chosen generator of the subgroup $\mathbb{G}_q$ of order q.

- $\hat{e}$ is an efficiently-computable bilinear pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, i.e., a map satisfying the following properties for $g \neq 1 \in \mathbb{G}$:
  - Bilinearity: $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$ for all $a, b \in \mathbb{Z}_{pq}$;
  - Non-degeneracy: $\hat{e}(g, g) \neq 1$.

Instead of setting the order of $\mathbb{G}$ to be the product of two primes (i.e. pq), we can set the order to be the product of multiple primes, e.g. [25, 26, 28]. In [28] and recapped in Section 3.2, the generator $\mathcal{G}_C{}^{(pqw)}$ generates $\mathbb{G}$ with the order of three primes (i.e. pqw).

Let $\Gamma = (p, q, \mathbb{G}, \mathbb{G}_T, \hat{e}, g_p, g_q)$ be the output by $\mathcal{G}_C{}^{(pq)}(\lambda)$, and $\Gamma^* = (pq, \mathbb{G}, \mathbb{G}_T, \hat{e}, g_p, g_q)$. We say the Composite-DDH assumption [4] holds if, for every P.P.T. attacker $\mathcal{A}$, its advantage $|\Pr[b' = b] - \frac{1}{2}|$ is negligible in the game, defined in Fig. 1.

```
1. $\Gamma = (p, q, \mathbb{G}, \mathbb{G}_T, \hat{e}, g_p, g_q) \xleftarrow{\$} \mathcal{G}_C{}^{(pq)}(\lambda)$
2. $a_1, a_2, b_1, b_2, b_3, r \xleftarrow{\$} \mathbb{Z}_{pq}$
3. $\Gamma^* = (pq, \mathbb{G}, \mathbb{G}_T, \hat{e}, g_p, g_q)$
4. $X_0 = (\Gamma^*, g_p^{a_1} \cdot g_q^{b_1}, g_p^{a_2} \cdot g_q^{b_2}, g_p^{a_1 a_2} \cdot g_q^{b_3})$
   $X_1 = (\Gamma^*, g_p^{a_1} \cdot g_q^{b_1}, g_p^{a_2} \cdot g_q^{b_2}, g_p^{r} \cdot g_q^{b_3})$
5. $b \xleftarrow{\$} \{0, 1\}$
6. $b' \xleftarrow{\$} \mathcal{A}(X_b)$
```

Fig. 1. Composite-DDH assumption

```
1. $\Gamma = (p, q, \mathbb{G}, \mathbb{G}_T, \hat{e}, g_p, g_q) \xleftarrow{\$} \mathcal{G}_C{}^{(pq)}(\lambda)$
2. $a_1, a_2, a_3, b_1, b_2, b_3, b_4, r \xleftarrow{\$} \mathbb{Z}_{pq}$
3. $\Gamma^* = (pq, \mathbb{G}, \mathbb{G}_T, \hat{e}, g_p, g_q)$
4. $X_0 = (\Gamma^*, g_p^{a_1} \cdot g_q^{b_1}, g_p^{a_2} \cdot g_q^{b_2}, g_p^{a_1 a_3} \cdot g_q^{b_3}, g_p^{a_2 a_3} \cdot g_q^{b_4})$
   $X_1 = (\Gamma^*, g_p^{a_1} \cdot g_q^{b_1}, g_p^{a_2} \cdot g_q^{b_2}, g_p^{a_1 a_3} \cdot g_q^{b_3}, g_p^{r} \cdot g_q^{b_4})$
5. $b \xleftarrow{\$} \{0, 1\}$
6. $b' \xleftarrow{\$} \mathcal{A}(X_b)$
```

Fig. 2. Weak Composite-DDH assumption

We say the Weak Composite-DDH assumption holds if, for every P.P.T. attacker $\mathcal{A}$, its advantage $|\Pr[b' = b] - \frac{1}{2}|$ is negligible in the game, defined in Fig. 2. Both assumptions are strictly weaker than the C3DH assumption by Boneh and Waters [12] because the attacker is given strictly more information in the C3DH attack game.

2.3 New Assumptions

In [15], Canetti proposed the DDH-II assumption which differs from the standard DDH assumption in that one exponent is chosen from a *wide spread* distribution instead of a uniform one. Damgård, Hazay and, Zottarel [18] showed that this assumption holds in the generic group model, and stated that it is a useful tool in leakage resilient cryptography.

Next, we introduce a new assumption for bilinear groups. The philosophy is similar to the case of Composite-DDH: although it is trivial to solve DDH-II problem in bilinear groups, adding an additional layer of randomization makes it difficult even with the help of the bilinear map. Formally, we say the Composite-DDH-II assumption holds if, for every P.P.T. attacker $\mathcal{A}$, its advantage $|\Pr[b' = b] - \frac{1}{2}|$ is negligible in the game, defined in Fig. 3. In the game, the distribution $\mathbb{D}$ from the attacker should guarantee that a_1 has min-entropy not smaller than λ, i.e. a_1 is *wide spread* according to Canetti [15].

```
1. $\Gamma = (p, q, \mathbb{G}, \mathbb{G}_T, \hat{e}, g, g_p, g_q) \xleftarrow{\$} \mathcal{G}_C{}^{(pq)}(\lambda)$
2. $\Gamma^* = (pq, \mathbb{G}, \mathbb{G}_T, \hat{e}, g, g_p, g_q)$
3. $\mathbb{D} \xleftarrow{\$} \mathcal{A}(\Gamma^*)$
4. $a_1 \xleftarrow{\mathbb{D}} \mathbb{Z}_p$
5. $b_1, s_1, s_2, s_3, r \xleftarrow{\$} \mathbb{Z}_{pq}$
6. $X_0 = (\Gamma^*, g_p^{a_1} \cdot g_q^{s_1}, g_p^{b_1} \cdot g_q^{s_2}, g_p^{a_1 b_1} \cdot g_q^{s_3})$
   $X_1 = (\Gamma^*, g_p^{a_1} \cdot g_q^{s_1}, g_p^{b_1} \cdot g_q^{s_2}, g_p^{r} \cdot g_q^{s_3})$
7. $b \xleftarrow{\$} \{0, 1\}$
8. $b' \xleftarrow{\$} \mathcal{A}(X_b, \mathbb{D})$
```

Fig. 3. Composite-DDH-II assumption

```
1. $\Gamma = (p, q, \mathbb{G}, \mathbb{G}_T, \hat{e}, g, g_p, g_q) \xleftarrow{\$} \mathcal{G}_C{}^{(pq)}(\lambda)$
2. $\Gamma^* = (pq, \mathbb{G}, \mathbb{G}_T, \hat{e}, g, g_p, g_q)$
3. $\mathbb{D} \xleftarrow{\$} \mathcal{A}(\Gamma^*)$
4. $(a_1, \cdots, a_L) \xleftarrow{\mathbb{D}} \mathbb{Z}_p^L$
5. $b_1, \cdots b_L, r_1, \cdots, r_L, s_1, \cdots, s_L, t_1, \cdots, t_L \xleftarrow{\$} \mathbb{Z}_{pq}$
6. $X_0 = (\Gamma^*, g_p^{b_1} \cdot g_q^{s_1}, g_p^{a_1 b_1} \cdot g_q^{t_1}, \cdots, g_p^{b_L} \cdot g_q^{s_L}, g_p^{a_L b_L} \cdot g_q^{t_L})$
   $X_1 = (\Gamma^*, g_p^{b_1} \cdot g_q^{s_1}, g_p^{r_1} \cdot g_q^{t_1}, \cdots, g_p^{b_L} \cdot g_q^{t_L}, g_p^{r_L} \cdot g_q^{t_L})$
7. $b \xleftarrow{\$} \{0, 1\}$
8. $b' \xleftarrow{\$} \mathcal{A}(X_b, \mathbb{D})$
```

Fig. 4. Correlated Composite-DDH-II assumption

Further, we say the Correlated Composite-DDH-II assumption holds if, for every P.P.T. attacker $\mathcal{A}$, its advantage $|\Pr[b' = b] - \frac{1}{2}|$ is negligible in the game, defined in Fig. 4. In the game, the distribution $\mathbb{D}$ from the attacker should guarantee that a_i for any $1 \leq i \leq L$ has min-entropy not smaller than λ. Compared to the Composite-DDH-II assumption, on one hand the values $g_p^{a_1}, \cdots, g_p^{a_L}$ are not given to the attacker (not even in the randomized form), but on the other hand the attacker is given multiple incomplete DH pairs. As to the two new assumptions, it is not clear how to reduce one to the other. Nevertheless, we have the following lemma with its proof in Appendix I.

Lemma 1. *Suppose any P.P.T. attacker has at most the advantage ϵ in the Composite-DDH-II assumption. Then, any P.P.T. attacker has at most the advantage $L \cdot \epsilon$ in the Correlated Composite-DDH-II assumption in the following two scenarios.*

1. *$a_1, a_2, \cdots, a_L$ are independent according to $\mathbb{D}$.*
2. *$a_1 = a_2 = \cdots = a_L$ according to $\mathbb{D}$.*

It is unclear how to reduce these two assumptions to existing standard assumptions. Nevertheless, we can prove their security in the generic group model, as what have been done for the DDH-II assumption in [18]. The proof will appear in the full paper.

3 Forward Security Properties for PEKS

A PEKS scheme involves a client, a server, and senders which can be any entity. Let λ be the security parameter, a PEKS scheme has the following algorithms.

- KeyGen(λ): Run by the client, this probabilistic algorithm outputs a public/private key pair (PK, SK), where PK should define a message space $\mathcal{W}$.

- $\mathsf{Encrypt}(x, PK)$: Run by a sender, this probabilistic algorithm outputs a ciphertext (or, tag) C_x for a message (or, keyword) $x \in \mathcal{W}$.
- $\mathsf{TrapGen}(y, SK)$: Run by the client, this probabilistic algorithm generates a trapdoor T_y for the message $y \in \mathcal{W}$.
- $\mathsf{Test}(C_x, T_y, PK)$: Run by the server, this deterministic algorithm returns 1 if $x = y$ and 0 otherwise.

Boneh et al. [8] defined ciphertext privacy property for PEKS, and Abdala et al. [1] defined computational consistency property. Next, we present the new forward security properties.

3.1 Forward Security Properties for PEKS

The forward-secure trapdoor unlinkability says that any P.P.T. attacker cannot determine the links among trapdoors as long as the underlying keywords are sampled according to distributions with min-entropy not smaller than λ. This property is an augmented variant of the strong search pattern privacy property from [4] in two aspects.

- The attacker is given SK in the attack game, and this brings in the forward security flavor.
- The attacker can adaptively specify the keyword distributions based on the public parameters, while the challenger samples the keywords uniformly from the keyword space in [4] (i.e. the attacker has no control on the keyword distributions).

Definition 1. *A PEKS scheme achieves forward-secure trapdoor unlinkability if any P.P.T. attacker $\mathcal{A}$'s advantage $|Pr[b' = b] - \frac{1}{2}|$ is negligible in the game shown in Fig. 5. In the game, $\mathbb{D}_0$ is the joint distribution of L (dependent) λ-source random variables, while $\mathbb{D}_1$ defines L independent random variables with uniform distribution. Chosen by the attacker, the integer L is a polynomial in λ.*

```
1. $(PK, SK) \xleftarrow{\$} \mathsf{KeyGen}(\lambda)$
2. $(\mathbb{D}_0, \mathbb{D}_1, L, state) \xleftarrow{\$} \mathcal{A}^{\mathsf{TrapGen}}(PK)$
3. $b \xleftarrow{\$} \{0,1\}$,  $\boldsymbol{x}_b \xleftarrow{\mathbb{D}_b} \mathcal{W}^L$,  $\boldsymbol{T}_b = \mathsf{TrapGen}(\boldsymbol{x}_b, SK)$
4. $b' \xleftarrow{\$} \mathcal{A}(\boxed{\mathrm{SK}}, state, \boldsymbol{T}_b)$
```

Fig. 5. Forward-Secure Trapdoor Unlinkability

```
1. $(PK, SK) \xleftarrow{\$} \mathsf{KeyGen}(\lambda)$
2. $(\mathbb{D}_0, \mathbb{D}_1, L, state) \xleftarrow{\$} \mathcal{A}^{\mathsf{TrapGen}}(PK)$
3. $b \xleftarrow{\$} \{0,1\}$,  $\boldsymbol{x}_b \xleftarrow{\mathbb{D}_b} \mathcal{W}^L$,  $\boldsymbol{T}_b = \mathsf{TrapGen}(\boldsymbol{x}_b, SK)$,  $\boldsymbol{C}_b = \mathsf{Encrypt}(\boldsymbol{x}_b, PK)$
4. $b' \xleftarrow{\$} \mathcal{A}^{\mathsf{TrapGen},\mathsf{Encrypt}}(\boxed{\mathrm{SK}}, state, \boldsymbol{T}_b, \boldsymbol{C}_b)$
```

Fig. 6. Forward-Secure Function Privacy

For simplicity, we use $\mathsf{TrapGen}(\boldsymbol{x}_b, SK)$ to denote $(\mathsf{TrapGen}(\boldsymbol{x}_b^{(1)}, SK), \cdots, \mathsf{TrapGen}(\boldsymbol{x}_b^{(L)}, SK))$ in Fig. 5. Such notation is also used in Fig. 6 and property definitions for IBE in Section 4.1.

The forward-secure function privacy property says that any P.P.T. attacker cannot determine the links among (tag, trapdoor) pairs, as long as the underlying keywords are sampled according to distributions with min-entropy not smaller than λ. This property is an augmented variant of the enhanced function privacy property from [9] in two aspects.

- The attacker is given SK in the attack game, and this brings in the forward security flavor.
- We get rid of the restriction in the enhanced function privacy property definition from [9], namely the conditional min-entropy of $x_0^{(i)}$ given $x_0^{(1)}, \cdots, x_0^{(i-1)}$ is required to be at least λ, for all $2 \le i \le L$.

Definition 2. *A PEKS scheme achieves* $\overline{\text{forward-secure function privacy}}$ *if any P.P.T. attacker* $\mathcal{A}$*'s advantage* $|Pr[b' = b] - \frac{1}{2}|$ *is negligible in the game shown in Fig. 6. In the game,* $\mathbb{D}_0$ *and* $\mathbb{D}_1$ *are defined in the same way as in Definition 1, but with the following restriction: for* $\mathbf{x}_0 = (x_0^{(1)}, x_0^{(2)}, \cdots, x_0^{(L)}) \xleftarrow{\mathbb{D}_0} \mathcal{W}^L$ *and any* $1 \le i \neq j \le L$*, the probability* $\Pr[x_0^{(i)} = x_0^{(j)}]$ *is negligible.*

In practice, the client may submit search queries for the same keyword multiple times and the pattern will be something as follows:

$$\underline{keyword1},\ keyword2,\ keyword3,\ \underline{keyword1},\ keyword4,\ keyword5,\ \underline{keyword1}, ...$$

However, the "Real-or-Random" definition approach does not allow us to *straightforwardly* capture this given that the attacker gets access to (tag, trapdoor) pairs. If we allow the sampled keywords to be equal according to $\mathbb{D}_0$, then an attacker can win the game trivially (by cross testing the trapdoors and the ciphertexts) because $\mathbb{D}_1$ samples the keywords uniformly at random. To bridge the gap, we give the attacker access to $\mathsf{TrapGen}$ and $\mathsf{Encrypt}$ oracles in Step 4 of the above game, to capture the fact that the attacker can access multiple trapdoors and tags for the same keywords. In a $\mathsf{TrapGen}$ oracle query, the attacker has an index $1 \le i \le L$ as input and receives $\mathsf{TrapGen}(x_b^{(i)}, SK)$. In an $\mathsf{Encrypt}$ oracle query, the attacker has an index $1 \le j \le L$ as input and receives $\mathsf{Encrypt}(x_b^{(j)}, PK)$.

Similar to the argument in [4], the forward-secure trapdoor unlinkability property and the forward-secure function privacy property do not imply each other. In comparison, the forward-secure trapdoor unlinkability property is stronger in the sense that $\mathbb{D}_0$ can allow identical random variables, while the forward-secure function privacy property is stronger in the sense that the attacker gets not only the trapdoors but also corresponding ciphertexts. We skip the details here.

3.2 Analysis of Nishioka Scheme

In [28], Nishioka modeled trapdoor unlinkability for a very restricted setting: the attacker is non-adaptive, the unlinkability is only for two trapdoors, and the model seems to be selective since the challenge keywords are chosen before the generation of other parameters (in the SPP experiment). We found a minor inconsistency in the original Nishioka scheme (referred to as Instance 3 in [28]), namely r_1 is defined to be $r_1 \xleftarrow{\$} \mathbb{Z}_p$ for the $\mathsf{TrapGen}$ algorithm but p is not included in the SK. There are two ways to get rid of this inconsistency.

- One is to include p in SK. This will make the scheme fail to achieve the forward-secure trapdoor unlinkability property even against RO-non-adaptive attackers.
- The other is to set $r_1 \xleftarrow{\$} \mathbb{Z}_{pqw}$, and the scheme works in the same way as in the case of $r_1 \xleftarrow{\$} \mathbb{Z}_p$. This leads to the description in Fig. 7.

$\underline{\mathsf{KeyGen}(\lambda)}$	$\underline{\mathsf{TrapGen}(y, SK)}$
$(p,q,w,\mathbb{G},\mathbb{G}_T,\hat{e},g_p,g_q,g_w) \xleftarrow{\$} \mathcal{G}_C^{(pqw)}(\lambda)$ $g_q^{\dagger} \xleftarrow{\$} \mathbb{G}_q,\ g = g_p \cdot g_q^{\dagger}$ $\mathcal{W} = \{0,1\}^*,\ \mathsf{H} : \{0,1\}^* \rightarrow \mathbb{G}_p$ $PK = (pqw, \mathbb{G}, \mathbb{G}_T, \hat{e}, g_q, g_w, g, \mathcal{W}, \mathsf{H})$ $SK = (PK, g_p)$	$\boxed{r_1 \xleftarrow{\$} \mathbb{Z}_{pqw}}$ $g'_w, g''_w \xleftarrow{\$} \mathbb{G}_w$ $T_1 = g_p^{r_1} \cdot g'_w$ $T_2 = \mathsf{H}(y)^{r_1} \cdot g''_w$ $T_y = (T_1, T_2)$
$\underline{\mathsf{Encrypt}(x, PK)}$	$\underline{\mathsf{Test}(C_x, T_y, PK)}$
$r_2 \xleftarrow{\$} \mathbb{Z}_{pqw},\ g'_q, g''_q \xleftarrow{\$} \mathbb{G}_q$ $C_1 = g^{r_2} \cdot g'_q,\ C_2 = \mathsf{H}(x)^{r_2} \cdot g''_q$ $C_x = (C_1, C_2)$	if $\hat{e}(T_1, C_2) = \hat{e}(T_2, C_1)$, output 1 otherwise, output 0

Fig. 7. Nishioka Scheme (with modification)

In Definition 1 and 2, we assume the attacker to be fully adaptive in the sense that it can choose the distribution $\mathbb{D}_0$ based on everything. An immediate relaxation on these definitions is to make the attacker *RO-non-adaptive*, which means that the attacker can choose the distribution $\mathbb{D}_0$ based on everything except for the random oracle (i.e. the hash function). In practice, the keywords in search queries might be related to the system parameters in some manner, but it is hard to imagine a scenario where the keywords would depend on the behavior of a random function. We argue that the relaxation is minimal and reasonable.

Following Theorem 6.1 from [10], based on the fact that the keywords are hashed in both the $\mathsf{TrapGen}$ and $\mathsf{Encrypt}$ algorithms, the scheme trivially achieves the forward-secure function privacy property in the random oracle model against RO-non-adaptive atatckers. However, it is not trivial for the forward-secure trapdoor unlinkability property, due to the fact that the attacker can let $\mathbb{D}_0$ output identical keywords and exploit this in the attack. We have the following theorem with its proof in Appendix II.

Theorem 1. *The scheme in Fig. 7 achieves the forward-secure trapdoor unlinkability property against RO-non-adaptive attackers based on the Weak Composite-DDH assumption in the random oracle model.*

Note that the Weak Composite-DDH assumption defined in Fig. 2 is for bilinear composite-order group of the order pq, in the above theorem we assume this assumption holds for any composite-order subgroup (i.e. $\mathbb{G}_{pq}$, $\mathbb{G}_{pw}$and $\mathbb{G}_{qw}$) of the bilinear group $\mathbb{G}$ with the order pqw.

4 IBE and its Security Properties

An IBE scheme is specified by four algorithms ($\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec}$), defined in Fig. 8. Let the message space be $\mathcal{M}$ and the identity space be $\mathcal{I}$. The generic transformation from IBE to PEKS, proposed in [1], works as in Fig. 9. Note that the message space $\mathcal{W}$ of the resulted PEKS scheme is the public-key space $\mathcal{I}$ of the IBE scheme.

1. $(Msk, params) = \mathsf{Setup}(\lambda)$
2. $sk_{id} = \mathsf{Extract}(Msk, id)$
3. $C = \mathsf{Enc}(m, id)$
4. $\mathsf{Dec}(C,\ sk_{id}) = m$ or $\perp$

Fig. 8. IBE

1. $\boxed{\mathsf{KeyGen}(\lambda)} = \mathsf{Setup}(\lambda)$
2. $\boxed{\mathsf{Encrypt}(x, PK)} = (m,\ \mathsf{Enc}(m, x))$, where $m \overset{\$}{\leftarrow} \mathcal{M}$
3. $\boxed{\mathsf{TrapGen}(y, SK)} = \mathsf{Extract}(Msk, y)$
4. $\boxed{\mathsf{Test}(C_x, T_y, PK)} = 1$ iff $m = \mathsf{Dec}(\mathsf{Enc}(m, x), T_y)$

Fig. 9. Resulted PEKS

The standard IND-CPA and anonymity properties for IBE can be found in [1], and we define two new forward security properties for IBE in the next subsection. Under our definitions, the generic transformation leads to the the following property mapping.

PEKS Properties	IBE Properties
computational consistency	IND-CPA
ciphertext privacy	anonymity
forward-secure trapdoor unlinkability	msk-forward-secure key unlinkability
forward-secure function privacy	msk-forward-secure function privacy

4.1 Forward Security Properties of IBE

The following msk-forward-secure key unlinkability property says that any P.P.T. attacker cannot determine the links among private keys if the underlying identities are sampled according to distributions with min-entropy not smaller than λ, even with the knowledge of the master secret key. This property is an augmented variant of the strong key unlinkability property from [4], where the augmentation lies in two aspects.

- The attacker is given Msk in the attack game, and this gives the forward security flavor.

- The attacker is allowed to adaptively choose the identity distribution $\mathbb{D}_0$ based on the public parameters, while the challenger samples the identities uniformly at random (according to certain patterns defined by the attacker) from the identity space in [4].

Definition 3. *An IBE scheme achieves* $\underline{\textsf{msk}\textit{-forward-secure key unlinkability}}$ *if any P.P.T. attacker* $\mathcal{A}$*'s advantage* $|Pr[b' = b] - \frac{1}{2}|$ *is negligible in the game shown in Fig. 10. In the game,* $\mathbb{D}_0$ *is the joint distribution of* L *(dependent)* λ*-source random variables, while* $\mathbb{D}_1$ *defines* L *independent random variables with uniform distribution. Chosen by the attacker, the integer* L *is a polynomial in* λ*.*

```
1. $(Msk, params) \xleftarrow{\$} \mathsf{Setup}(\lambda)$
2. $(\mathbb{D}_0, \mathbb{D}_1, L, state) \xleftarrow{\$} \mathcal{A}^{\mathsf{Extract}}(params)$
3. $b \xleftarrow{\$} \{0,1\}$, $\boldsymbol{id}_b \xleftarrow{\mathbb{D}_b} \mathcal{I}^L$, $\boldsymbol{sk}_b = \mathsf{Extract}(Msk, \boldsymbol{id}_b)$
4. $b' \xleftarrow{\$} \mathcal{A}(\boxed{\text{Msk}}, state, \boldsymbol{sk}_b)$
```

Fig. 10. msk-Forward-Secure Key Unlinkability

```
1. $(PK, SK) \xleftarrow{\$} \mathsf{Setup}(\lambda)$
2. $(\mathbb{D}_0, \mathbb{D}_1, L, state) \xleftarrow{\$} \mathcal{A}^{\mathsf{Extract}}(params)$
3. $b \xleftarrow{\$} \{0,1\}$, $\boldsymbol{id}_b \xleftarrow{\mathbb{D}_b} \mathcal{I}^L$, $\boldsymbol{sk}_b = \mathsf{Extract}(Msk, \boldsymbol{id}_b)$,
   $\boldsymbol{m}_b \xleftarrow{\$} \mathcal{M}^L$, $\boldsymbol{C}_b = \mathsf{Enc}(\boldsymbol{m}_b, \boldsymbol{id}_b)$
4. $b' \xleftarrow{\$} \mathcal{A}^{\mathsf{Extract,Enc}}(\boxed{\text{Msk}}, state, \boldsymbol{sk}_b, \boldsymbol{C}_b)$
```

Fig. 11. msk-Forward-Secure Function Privacy

The following msk-forward-secure function privacy property says that any P.P.T. attacker cannot determine the links among (private key, ciphertext) pairs if the underlying identities are sampled according to distributions with min-entropy not smaller than λ, even with the knowledge of the master secret key. This property is an augmented variant of the enhanced function privacy property from [9], where the augmentation lies in two aspects.

- The attacker is given *Msk* in the attack game, and this gives the forward security flavor.
- We get rid of this restriction in the enhanced function privacy property definition from [9], namely the conditional min-entropy of $\boldsymbol{id}_0^{(i)}$ given $\boldsymbol{id}_0^{(1)}, \cdots, \boldsymbol{id}_0^{(i-1)}$ is required to be at least λ, for all $2 \leq i \leq L$.

Definition 4. *An IBE scheme achieves* $\underline{\textsf{msk}\textit{-forward-secure function privacy}}$ *if any P.P.T. attacker* $\mathcal{A}$*'s advantage* $|Pr[b' = b] - \frac{1}{2}|$ *is negligible in the game shown in Fig. 11. In the game,* $\mathbb{D}_0$ *and* $\mathbb{D}_1$ *are defined in the same way as in Definition 3, but with the following restriction: for* $\boldsymbol{id}_0 = (\boldsymbol{id}_0^{(1)}, \boldsymbol{id}_0^{(2)}, \cdots, \boldsymbol{id}_0^{(L)}) \xleftarrow{\mathbb{D}_0} \mathcal{I}^L$ *and any* $1 \leq i \neq j \leq L$*, the probability* $\Pr[\boldsymbol{id}_0^{(i)} = \boldsymbol{id}_0^{(j)}]$ *is negligible.*

In Step 3 of the attack game, we use $\mathsf{Enc}(\boldsymbol{m}_b, \boldsymbol{id}_b)$ to denote $(\mathsf{Enc}(\boldsymbol{m}_b^{(1)}, \boldsymbol{id}_b^{(1)}), \cdots, \mathsf{Enc}(\boldsymbol{m}_b^{(L)}, \boldsymbol{id}_b)^{(L)})$ for the simplicity of notation. For the same reason as that in the definition of forward-secure function privacy for PEKS (i.e. Definition 2), the attacker is given access to the Extract and Enc oracles in Step 4 of the above game. In an Extract oracle query, the attacker has an index $1 \leq i \leq L$ as input and receives $\mathsf{Extract}(Msk, \boldsymbol{id}_b^{(i)})$. In an Enc oracle the attacker has an index $1 \leq j \leq L$ as input and receives $\mathsf{Enc}(m, \boldsymbol{id}_b^{(j)})$.

5 Analysis of Two Existing IBE Schemes

In this section, we analyse an IBE scheme by Boneh et al. [10] and an IBE scheme by Arriaga et al. [4] in our security model.

5.1 Boneh-Raghunathan-Segev $\mathcal{IBE}_{\mathsf{DLIN2}}$ Scheme

$\underline{\mathsf{Setup}(\lambda)}$	$\underline{\mathsf{Extract}(Msk, id)}$
$\Gamma = (\mathbb{G}, \mathbb{G}_T, \hat{e}, g, p) = \mathcal{G}_{\mathcal{P}}(\lambda)$ $\boldsymbol{A}_0, \boldsymbol{B}, \boldsymbol{A}_1, \cdots, \boldsymbol{A}_n \xleftarrow{\$} \mathbb{Z}_p^{2\times m}$ $\boldsymbol{u} \xleftarrow{\$} \mathbb{Z}_p^2$, $\mathcal{M} = \mathbb{G}_T$, $\mathcal{I} = \{0,1\}^n$ $Msk = (\boldsymbol{A}_0, \boldsymbol{B}, \boldsymbol{A}_1, \cdots, \boldsymbol{A}_n, \boldsymbol{u})$ $params = (\Gamma, g^{\boldsymbol{A}_0}, \boldsymbol{B}, g^{\boldsymbol{A}_1}, \cdots, g^{\boldsymbol{A}_n}, g^{\boldsymbol{u}}, \mathcal{M}, \mathcal{I})$	$id = (id_1, id_2, \cdots, id_n) \in \{0,1\}^n$ $\boldsymbol{S} \xleftarrow{\$} \mathbb{Z}_p^{m\times 2}$ $\boldsymbol{F}_{id,S} = [\boldsymbol{A}_0 \vert \boldsymbol{BS} + (\sum_{1\le j\le n} id_j \boldsymbol{A}_j)\boldsymbol{S}]$ $\boldsymbol{v} \xleftarrow{\$} \{\boldsymbol{x} \mid \boldsymbol{F}_{id,S} \cdot \boldsymbol{x} = \boldsymbol{u} \pmod p\}$ $\boldsymbol{z} = g^{\boldsymbol{v}} \in \mathbb{G}^{m+2}$, $sk_{id} = (\boldsymbol{S}, \boldsymbol{z})$
$\underline{\mathsf{Enc}(m, id)}$	$\underline{\mathsf{Dec}(C, sk_{id})}$
$id = (id_1, id_2, \cdots, id_n) \in \{0,1\}^n$, $m \in \mathbb{G}_T$ $\boldsymbol{D}(id) = \sum_{1\le j\le n} id_j \boldsymbol{A_j}$, $\boldsymbol{r} \xleftarrow{\$} \mathbb{Z}_p^2$ $\boldsymbol{c_0}^T = g^{\boldsymbol{r}^T \boldsymbol{A_0}}$, $\boldsymbol{c_1}^T = g^{\boldsymbol{r}^T[\boldsymbol{B}+\boldsymbol{D}(id)]}$, $c_2 = m \cdot \hat{e}(g,g)^{\boldsymbol{r}^T \boldsymbol{u}}$ $C = (\boldsymbol{c_0}, \boldsymbol{c_1}, c_2)$	$\boldsymbol{d}^T = [\boldsymbol{c_0}^T \vert (\boldsymbol{c_1}^T)^S] = g^{\boldsymbol{r}^T \boldsymbol{F}_{id,S}}$ $\hat{e}(\boldsymbol{d}, \boldsymbol{z}) = \hat{e}(g,g)^{\boldsymbol{r}^T(\boldsymbol{F}_{id,S}\cdot \boldsymbol{v})} = \hat{e}(g,g)^{\boldsymbol{r}^T \boldsymbol{u}}$ $m = c_2 \cdot \hat{e}(\boldsymbol{d}, \boldsymbol{z})^{-1}$

Fig. 12. Boneh-Raghunathan-Segev $\mathcal{IBE}_{\mathsf{DLIN2}}$ Scheme

According to their definitions, the $\mathcal{IBE}_{\mathsf{DLIN2}}$ scheme achieves enhanced function privacy based on the DLIN assumption in the standard model. Next, we show that this scheme does not achieve msk-forward-secure function privacy, namely an attacker wins the attack game in Fig. 11 with overwhelming probability. Note that this does not conflict with the claims from [10] because our security model is stronger. The following attack makes use of the fact that, with *Msk*, the attacker can transform a ciphertext under an identity *id* into a ciphertext under another identity *id′*, for some carefully chosen *id* and *id′*. To mount an attack, in step 2 and 4 of the game, the attacker performs as follows.

- In step 2, the attacker sets $L = 2$, which means $\mathbb{D}_0$ and $\mathbb{D}_1$ are the joint distribution of two identity variables. The attacker defines $\mathbb{D}_0$ as follows: $\boldsymbol{id}_0^{(1)} = (id_1, id_2, \cdots, id_n)$ is defined as $(id_1, id_2, \cdots, id_{n-1}) \xleftarrow{\$} \{0,1\}^{n-1}$ and $id_n = 0$; $\boldsymbol{id}_0^{(2)}$ equals $\boldsymbol{id}_0^{(1)}$ except its $id_n = 1$.
- In step 4, the attacker firstly obtains $Msk = (\boldsymbol{A}_0, \boldsymbol{B}, \boldsymbol{A}_1, \cdots, \boldsymbol{A}_n, \boldsymbol{u})$. Then, the attacker computes $\boldsymbol{X} \in \mathbb{Z}_p^{m\times m}$ such that $\boldsymbol{A_n} = \boldsymbol{A_0}\boldsymbol{X}$. Recall that the challenge is $(\boldsymbol{sk}_b, \boldsymbol{C}_b)$. The first ciphertext in $\boldsymbol{C}_b$, namely $\boldsymbol{C}_b^{(1)} = (\boldsymbol{c_0}, \boldsymbol{c_1}, c_2)$, is in the

following form: $\boldsymbol{c_0}^T = g^{\boldsymbol{r}^T \boldsymbol{A_0}}, \boldsymbol{c_1}^T = g^{\boldsymbol{r}^T [\boldsymbol{B}+\boldsymbol{D}(\boldsymbol{id}_b^{(1)})]}, c_2 = m \cdot \hat{e}(g,g)^{\boldsymbol{r}^T \boldsymbol{u}}$. The attacker has a new ciphertext $C' = (\boldsymbol{c_0}, \boldsymbol{c'_1}, c_2)$, where

$$\begin{aligned} \boldsymbol{c'_1}^T &= \boldsymbol{c_1}^T \otimes (\boldsymbol{c_0}^T)^{\boldsymbol{X}} \\ &= g^{\boldsymbol{r}^T [\boldsymbol{B}+\boldsymbol{D}(\boldsymbol{id}_b^{(1)})]} \otimes g^{\boldsymbol{r}^T \boldsymbol{A_0} \boldsymbol{X}} \\ &= g^{\boldsymbol{r}^T [\boldsymbol{B}+\boldsymbol{D}(\boldsymbol{id}_b^{(1)})]} \otimes g^{\boldsymbol{r}^T \boldsymbol{A_n}} \end{aligned}$$

Let the secret keys in the challenge $\boldsymbol{sk}_b$ be denoted as $(sk_{\boldsymbol{id}_b^{(1)}}, sk_{\boldsymbol{id}_b^{(2)}})$. The attacker outputs 0 if $\mathsf{Dec}(C', sk_{\boldsymbol{id}_b^{(2)}}) = \mathsf{Dec}(\boldsymbol{C}_b^{(1)}, sk_{\boldsymbol{id}_b^{(1)}})$, and outputs 1 otherwise.

Recall that, $\otimes$ is an operator for pairwise group operations between two the new vectors or matrices. It is clear that if $b = 0$ then we have $\boldsymbol{c'_1}^T = g^{\boldsymbol{r}^T [\boldsymbol{B}+\boldsymbol{D}(\boldsymbol{id}_b^{(2)})]}$ and the equality $\mathsf{Dec}(\boldsymbol{C}_b^{(1)}, sk_{\boldsymbol{id}_b^{(1)}}) = \mathsf{Dec}(C', sk_{\boldsymbol{id}_b^{(2)}})$ holds. But, this equality holds with a negligible probability if $b = 1$. As a result, our attack works.

5.2 Arriaga-Tang-Ryan IBE Scheme

The following scheme was proposed by Arriaga et al. [4], based on an anonymous IBE scheme by Boyen and Waters [13]. This scheme has been proven secure with respect to the strong key unlinkability property (under the definition in [4]) in the random oracle model. Compared with our msk-forward-secure key unlinkability property, their definition is weaker in three aspects: (1) the attacker is not allowed to adaptively choose the identity distribution $\mathbb{D}_0$ and it can only specify the identity patterns (i.e. which identities are equal); (2) according to the patterns, the challenger samples the identities uniformly at random from the identity space; (3) there is no forward security.

$\underline{\mathsf{Setup}(\lambda)}$	$\underline{\mathsf{Extract}(Msk, id)}$
$\Gamma = (p, q, \mathbb{G}, \mathbb{G}_T, \hat{e}, g, g_p, g_q) = \mathcal{G}_C(\lambda)$ $\Gamma^* = (n = pq, \mathbb{G}, \mathbb{G}_T, \hat{e}, g, g_p, g_q)$ $x, t_1, t_2 \xleftarrow{\$} \mathbb{Z}_n$ $\Omega = \hat{e}(g_p, g_p)^{xt_1t_2}, v_1 = g_p^{t_1}, v_2 = g_p^{t_2}$ $\mathcal{M} = \mathbb{G}_T, \mathcal{I} = \{0,1\}^*, \mathsf{H} : \{0,1\}^* \to \mathbb{G}_p$ $Msk = (x, t_1, t_2), params = (\Gamma^*, \Omega, v_1, v_2, \mathcal{M}, \mathcal{I}, \mathsf{H})$	$r \xleftarrow{\$} \mathbb{Z}_n$ $x_0, x_1, x_2 \xleftarrow{\$} \mathbb{G}_q$ $d_0 = x_0 \cdot g_p^{rt_1t_2}$ $d_1 = x_1 \cdot g_p^{-xt_2} \cdot \mathsf{H}(id)^{-rt_2}$ $d_2 = x_2 \cdot g_p^{-xt_1} \cdot \mathsf{H}(id)^{-rt_1}$ $sk_{id} = (d_0, d_1, d_2)$
$\underline{\mathsf{Enc}(m, id)}$	$\underline{\mathsf{Dec}(C, sk_{id})}$
$s, s_1 \xleftarrow{\$} \mathbb{Z}_n$ $\hat{c} = \Omega^s m$, $c_0 = \mathsf{H}(id)^s$, $c_1 = v_1^{s-s_1}$, and $c_2 = v_2^{s_1}$ $C = (\hat{c}, c_0, c_1, c_2)$	$e_0 = \hat{e}(c_0, d_0), e_1 = \hat{e}(c_1, d_1)$ $e_2 = \hat{e}(c_2, d_2)$, $m = \hat{c} \cdot e_0 \cdot e_1 \cdot e_2$

Fig. 13. Arriaga-Tang-Ryan IBE Scheme

Similar to the discussions in Section 3.2, an immediate relaxation on Definition 3 and 4 is to make the attacker *RO-non-adaptive*, which means that the attacker can choose the distribution $\mathbb{D}_0$ based on everything except for the random oracle (i.e. the hash function). Following Theorem 6.1 from [10], it is trivial to show that the scheme achieves msk-forward-secure function privacy property in the random oracle model. However, it is non-trivial for the msk-forward-secure key unlinkability property. We have the following theorem with its proof in Appendix III. It is worth noting that this result is stronger than Lemma 3 from [4] while it relies on a weaker assumption (i.e. Weak Composite-DDH assumption instead of Composite-DDH assumption).

Theorem 2. *The scheme achieves the* msk*-forward-secure key unlinkability property against RO-non-adaptive attackers based on the Weak Composite-DDH assumption in the random oracle model.*

6 msk-Forward-Secure IBE Construction

The "extract-combine-augment" concept from [9] is an elegant idea, but it has two drawbacks. One is that it introduces the unrealistic conditional min-entropy restriction on identity distribution when defining enhanced function privacy. The other is that it requires specific modifications to both encryption and decryption algorithms of the underlying IBE scheme. Such modifications may not be an easy task. Moreover, it may introduce *good* algebraic structures into the ciphertexts. This partially makes it possible for us to show that the $\mathcal{IBE}_{\mathsf{DLIN2}}$ scheme does not achieve msk-forward-secure function privacy in Section 5.1.

In the following, we first introduce the concept of correlated-input indistinguishable hash function, which serves as a building block to pre-process identities for any IBE scheme. Similar to the "extract" step in the "extract-combine-augment" approach, such a hash function aims at eliminating the correlations among different inputs so that msk-forward-secure function privacy can be straightforwardly achieved. The advantage is that there is no need to modify the underlying IBE algorithms. We then take the Boyen-Waters scheme [13] as an example to show how to make it msk-forward-secure.

6.1 Correlated-Input Indistinguishable Hash Function

Goyal et al. [21] introduced the concept of correlated-input secure hash functions and gave a few security definitions and instantiations. Unfortunately, their security property definitions are selective and assume certain specific correlations among the inputs (i.e. the inputs are related by polynomials over the input space). Such restrictions conflict with our needs, because we want the inputs to be arbitrarily correlated and full security. Moreover, we want a property which is subtly different from correlated-input pseudorandomness. Very informally, the pseudorandomness property guarantees that the outputs of a hash function look random with respect to correlated inputs, while our desired property is supposed to guarantee that the outputs of a hash function look the same with respect to correlated inputs and random inputs. Formally, we define the new property as follows.

Definition 5. *A hash function* $\mathsf{H} : \mathcal{X} \rightarrow \mathcal{Y}$ *is correlated-input indistinguishable if the attacker's advantage* $|\Pr[b' = b] - \frac{1}{2}|$ *is negligible in the attack game, shown in Fig. 14. In the game,* $\mathbb{D}_0$ *is the joint distributions of* L *(dependent)* λ*-source random variables over* $\mathcal{X}$*, while* $\mathbb{D}_1$ *defines* L *independent random variables with uniform distribution over* $\mathcal{X}$*. It is required that, for* $(\boldsymbol{x}_0^{(1)}, \boldsymbol{x}_0^{(2)}, \cdots, \boldsymbol{x}_0^{(L)}) \xleftarrow{\mathbb{D}_0} \mathcal{X}^L$ *and any* $1 \leq i \neq j \leq L$*, the probability* $\Pr[\boldsymbol{x}_0^{(i)} = \boldsymbol{x}_0^{(j)}]$ *is negligible. Chosen by the attacker, the integer* L *is a polynomial of the security parameter.*

```
1. $(\mathbb{D}_0, \mathbb{D}_1, L, state) \xleftarrow{\$} \mathcal{A}(\mathsf{H})$
2. $b \xleftarrow{\$} \{0,1\}$, $\boldsymbol{x}_b \xleftarrow{\mathbb{D}_b} \mathcal{X}^L$, $\boldsymbol{y}_b = (\mathsf{H}(\boldsymbol{x}_b^{(1)}), \cdots, \mathsf{H}(\boldsymbol{x}_b^{(L)}))$
3. $b' \xleftarrow{\$} \mathcal{A}(state, \boldsymbol{y}_b)$
```

Fig. 14. Correlated-Input Indistinguishability

Due to the different security objectives, it is easy to verify that the construction from [21] is not correlated-input indistinguishable. Bellare, Hoang, and Keelveedhi [6] introduced the concept of Universal Computational Extractors(UCEs) and showed how to use this concept to construct selective correlated-input secure hash functions according to the definitions from [21]. However, they noted that UCEs do not guarantee adaptive/full security. It seems difficult to construct correlated-input indistinguishable hash functions based on UCEs. Unseeded deterministic extractors also seem to be a related primitive, but the existing security models do not take into account correlated inputs.

On the positive side, we can instantiate correlated-input indistinguishable hash function based on deterministic encryption (DE) schemes, a primitive proposed in [5]. More specifically, the instantiation should be based on adaptively secure DE schemes, e.g. that from [29]. In a nutshell, an adaptively secure DE scheme guarantees that an attacker cannot distinguish the ciphertexts of arbitrarily correlated plaintexts and random plaintexts. The instantiation has two steps: (1) given an input from domain $\mathcal{X}$, encrypt it with the DE scheme to obtain a ciphertext; (2) hash the ciphertext with a collision-resistant hash function to get an output for the domain $\mathcal{Y}$.

6.2 Example msk-Forward-Secure IBE Construction

With a correlated-input indistinguishable hash function H, we describe an extended variant for the Boyen-Waters scheme [13] in Fig. 15. The extension is from two aspects: (1) pre-processing identities with H; (2) employ composite-order bilinear groups.

$\underline{\mathsf{Setup}(\lambda)}$	$\underline{\mathsf{Extract}(Msk, id)}$
$\Gamma = (p, q, \mathbb{G}, \mathbb{G}_T, \hat{e}, g_p, g_q) = \mathcal{G}_C(\lambda)$	$r_1, r_2 \xleftarrow{\$} \mathbb{Z}_n,$
$\Gamma^* = (n = pq, \mathbb{G}, \mathbb{G}_T, \hat{e}, g_p, g_q)$	$x_0, x_1, x_2, x_3, x_4 \xleftarrow{\$} \mathbb{G}_q$
$x, t_1, t_2, t_3, t_4 \xleftarrow{\$} \mathbb{Z}_n$	$d_0 = x_0 \cdot g_p^{r_1 t_1 t_2 + r_2 t_3 t_4}$
$\Omega = \hat{e}(g_p, g_p)^{x t_1 t_2}, g_0, g_1 \xleftarrow{\$} \mathbb{G}_p$	$d_1 = x_1 \cdot g_p^{-x t_2} \cdot (g_0 g_1^{\mathsf{H}(id)})^{-r_1 t_2}$
$v_1 = g_p^{t_1}, v_2 = g_p^{t_2}, v_3 = g_p^{t_3}, v_4 = g_p^{t_4}$	$d_2 = x_2 \cdot g_p^{-x t_1} \cdot (g_0 g_1^{\mathsf{H}(id)})^{-r_1 t_1}$
$\mathcal{M} = \mathbb{G}_T, \mathcal{I} = \mathbb{Z}_n, \mathsf{H} : \mathcal{I} \to \mathcal{I}$	$d_3 = x_3 \cdot (g_0 g_1^{\mathsf{H}(id)})^{-r_2 t_4}$
$Msk = (x, t_1, t_2, t_3, t_4)$	$d_4 = x_4 \cdot (g_0 g_1^{\mathsf{H}(id)})^{-r_2 t_3}$
$params = (\Gamma^*, g_0, g_1, \Omega, v_1, v_2, v_3, v_4, \mathcal{M}, \mathcal{I}, \mathsf{H})$	$sk_{id} = (d_0, d_1, d_2, d_3, d_4)$
$\underline{\mathsf{Enc}(m, id)}$	$\underline{\mathsf{Dec}(C, sk_{id})}$
$s, s_1, s_2 \xleftarrow{\$} \mathbb{Z}_n$	$e_0 = \hat{e}(c_0, d_0), e_1 = \hat{e}(c_1, d_1)$
$\hat{c} = \Omega^s m, c_0 = (g_0 g_1^{\mathsf{H}(id)})^s, c_1 = v_1^{s - s_1}$	$e_2 = \hat{e}(c_2, d_2), e_3 = \hat{e}(c_3, d_3)$
$c_2 = v_2^{s_1}, c_3 = v_3^{s - s_2}, c_4 = v_4^{s_2}$	$e_4 = \hat{e}(c_4, d_4)$
$C = (\hat{c}, c_0, c_1, c_2, c_3, c_4)$	$m = \hat{c} \cdot e_0 \cdot e_1 \cdot e_2 \cdot e_3 \cdot e_4$

Fig. 15. Extended Boyen-Waters Scheme

If H is correlated-input indistinguishable and collision-resistant, it is straightforward to verify that the extended scheme is IND-CPA, anonymous, and achieves msk-forward-secure function privacy. Next, we prove the scheme also achieves msk-forward-secure key unlinkability. The proof appears in Appendix IV.

Theorem 3. *The extended Boyen-Waters scheme achieves* msk*-forward-secure key unlinkability based on the correlated Composite-DDH-II assumption, given that* H *is correlated-input indistinguishable and collision-resistant.*

Our extension is similar to the Arriaga-Tang-Ryan IBE Scheme recapped in Section 5.2, but it does not simplify the original scheme so that it is possible to have provable security in the standard model (depending on H).

7 Related Work

In the seminal definition [8], PEKS only supports equality testing of keywords. To support more types of search queries, a number of extensions have been proposed. Among them, [12, 23, 24] support search queries with conjunctive keywords, [12, 31] support subset and range queries, and [25] supports disjunctions, polynomial equations, and inner products. In contrast to the large number of follow-up works to extend the PEKS functionality, very little has been done to investigate its full security capabilities and the only few we know are [4, 9, 10, 28, 30], where [30] only aims at a very restricted setting (i.e. with designated tester).

7.1 Concerning the Entropy of Keywords

With respect to PEKS and its extensions, there is a concern about the low entropy nature of keywords. For example, in the aforementioned email routing example, the entropy of keywords may not be very high. In [14], Byun et al. described offline keyword guessing attacks. This makes people wonder the practicality of privacy properties for PKES, particularly the function privacy property [10] and the search pattern privacy property [4]. Nevertheless, we would like to argue that it still makes sense to investigate the maximal level of security guarantees by PEKS. Theoretically, it is always interesting to study the strongest security properties for a cryptographic primitive. This has been done for many other primitives, such as encryption and signature schemes. Practically, it is not true that the keywords always have low entropy. When a PEKS scheme is deployed, the underlying application can enrich the keyword set with some context information. For instance, instead of using the keyword "confidential", the application can use "confidential:project-peks". A more effective way to augment the entropy is using pre-shared passwords between message senders and the client. An extra advantage of this approach is that there is no need to store passwords given they are memorable. It is worth noting that augmenting the entropy of keywords may cause some efficiency issues (e.g. there may be several augmented keywords for the same keyword "confidential", so that the client needs to generate several trapdoors to search for all confidential emails). This can be regarded as a natural tradeoff between security and efficiency.

7.2 Related Forward Security and Key Escrow Notions

Forward secrecy is a well-known property for key agreement protocols [19, 22], and it ensures that a session key will remain secure even if one of the long-term secret keys is compromised in the future. This concept has also been applied to other primitives, such as signature schemes [3, 7] and hierarchical identity based encryption (HIBE) schemes [16, 27, 32]. It is worth noting that the adapted forward secrecy notions in [3, 7, 16, 27, 32] focus on the key evolution problem: *the life cycle of the secret key is divided into n time slots (e.g. $t_0, t_1, \cdots, t_n$) and sk_i will evolve to $sk_{i+1} = \mathsf{f}(sk_i)$ based on a function* f *from time t_i to t_{i+1}; the forward secrecy properties guarantee that if sk_{i+1} is compromised then the operations done with $sk_0, \cdots, sk_i$ remain secure*. In the case of HIBE schemes [16, 27, 32], the focus is on the secret keys for certain identities instead of the master secret key. The forward security properties, introduced in Section 4, stem from [19, 22], and differs from that [16, 27, 32] in two aspects: the attacker has access to the master secret key and no key evolution is considered.

For identity-based cryptography, how to avoid the key escrow problem has been an interesting question, see e.g. [2, 20]. Among all, a particularly interesting security notion is the anonymous ciphertext indistinguishability (ACI) property from [17]. The ACI property guarantees that an attacker cannot determine the public key (or, identity) behind a ciphertext even with the knowledge of the master secret key. It is related to the enhanced function privacy property from [9], but they are not comparable: the ACI property is stronger in the sense that the attacker knows the master secret key, but it is weaker in the sense that it only considers a single uniformly chosen identity while the enhanced function privacy property considers a sequence of non-uniformly chosen and correlated identities.

8 Concluding Remarks

In this paper, we have defined some forward security properties for PEKS and IBE respectively. We have also analyzed several existing PEKS and IBE schemes, and extended the Boyen-Waters anonymous IBE scheme by using a new building block (i.e. correlated-input indistinguishable hash function). Our analysis shows that it is relatively easy to achieve our properties against RO-non-adaptive attackers while it is quite hard to construct secure schemes against adaptive attackers (in particular in the standard model). Our work has motivated many interesting future directions. As to the concept of correlated-input indistinguishable hash function, we only know one method to instantiate it. It is a very interesting task to construct correlated-input indistinguishable hash functions in other ways. As to the "extract-combine-augment" approach from [9], it is very elegant albeit it does not guarantee any msk-forward security properties. It is an interesting task to augment the concept and the schemes (e.g. the $\mathcal{IBE}_{\mathsf{DLIN2}}$ scheme) to achieve our msk-forward security properties. Both PEKS and IBE are special types of functional encryption [11]. Hence, the concept of forward security is also valuable for other functional encryption schemes, including other PEKS variants and searchable encryption schemes in the symmetric-key setting. This is a widely open research area for the future.

References

1. M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions. *J. Cryptol.*, 21(3):350–391, 2008.
2. S. S. Al-Riyami and K. G. Paterson. Certificateless public key cryptography. In C. Laih, editor, *Advances in Cryptology – ASIACRYPT 2003*, volume 2894 of *LNCS*, pages 452–473, 2003.
3. R. Anderson. Two remarks on public key cryptology. Technical Report UCAM-CL-TR-549, Cambridge University, 1997.
4. A. Arriaga, Q. Tang, and P. Ryan. Trapdoor privacy in asymmetric searchable encryption schemes. In D. Pointcheval and D. Vergnaud, editors, *Progress in Cryptology – AFRICACRYPT 2014*, volume 8469 of *LNCS*, pages 31–50. Springer, 2014.
5. M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In A. Menezes, editor, *Advances in cryptology — CRYPTO 2007*, volume 4622 of *LNCS*, pages 535–552. Springer, 2007.
6. M. Bellare, V. T. Hoang, and S. Keelveedhi. Instantiating random oracles via uces. In R. Canetti and J. A. Garay, editors, *Advances in Cryptology - CRYPTO 2013*, volume 8043 of *LNCS*, pages 398–415. Springer, 2013.
7. M. Bellare and S. K. Miner. A forward-secure digital signature scheme. In M. J. Wiener, editor, *Advances in Cryptology — CRYPTO 1999*, volume 1666 of *LNCS*, pages 431–448. Springer, 1999.

8. D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public Key Encryption with Keyword Search. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology — EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 506–522. Springer, 2004.
9. D. Boneh, A. Raghunathan, and G. Segev. Function-private identity-based encryption: Hiding the function in functional encryption. In R. Canetti and J. A. Garay, editors, *Advances in Cryptology — CRYPTO 2013*, volume 8043 of *LNCS*, pages 461–478. Springer, 2013.
10. D. Boneh, A. Raghunathan, and G. Segev. Function-private identity-based encryption: Hiding the function in functional encryption. http://eprint.iacr.org/2013/283.pdf, 2013.
11. D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In Y. Ishai, editor, *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011*, volume 6597 of *LNCS*, pages 253–273. Springer, 2011.
12. D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In S. P. Vadhan, editor, *Proceedings of the 4th conference on Theory of cryptography*, volume 4392 of *LNCS*, pages 535–554. Springer, 2007.
13. X. Boyen and B. Waters. Anonymous hierarchical identity-based encryption (without random oracles). In C. Dwork, editor, *Advances in Cryptology — CRYPTO 2006*, volume 4117 of *LNCS*, pages 290–307. Springer, 2006.
14. J. W. Byun, H. S. Rhee, H.Park, and D. H. Lee. Off-Line Keyword Guessing Attacks on Recent Keyword Search Schemes over Encrypted Data. In W. Jonker and M. Petkovic, editors, *Secure Data Management, Third VLDB Workshop, SDM 2006*, volume 4165 of *LNCS*, pages 75–83. Springer, 2006.
15. R. Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In B. S. Kaliski Jr., editor, *Advances in Cryptology — CRYPTO 1997*, volume 1294 of *Lecture Notes in Computer Science*, pages 455–469. Springer, 1997.
16. R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. In E. Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 255–271. Springer, 2003.
17. S. M. Chow. Removing escrow from identity-based encryption. In S. Jarecki and G. Tsudik, editors, *Public Key Cryptography – PKC 2009, 12th International Conference on Practice and Theory in Public Key Cryptography*, volume 5443 of *LNCS*, pages 256–276. Springer, 2009.
18. I. Damgård, C. Hazay, and A. Zottarel. Short paper on the generic hardness of DDH-II. http://cs.au.dk/ angela/Hardness.pdf, Accessed in May, 2014.
19. W. Diffie, P. C. Oorschot, and M. J. Wiener. Authentication and authenticated key exchanges. *Designs, Codes and Cryptography*, 2(2):107–125, 1992.
20. V. Goyal. Reducing trust in the PKG in identity based cryptosystems. In A. Menezes, editor, *Advances in Cryptology – CRYPTO 2007*, volume 4622 of *LNCS*, pages 430–447. Springer, 2007.
21. V. Goyal, A. O'Neill, and V. Rao. Correlated-input secure hash functions. In Y. Ishai, editor, *Theory of Cryptography — 8th Theory of Cryptography Conference, TCC 2011*, volume 6597 of *Lecture Notes in Computer Science*, pages 182–200. Springer, 2011.
22. C. Günther. An identity-based key-exchange protocol. In J. Quisquater and J. Vandewalle, editors, *Advances in Cryptology — EUROCRYPT 1989*, volume 434 of *Lecture Notes in Computer Science*, pages 29–37. Springer, 1990.
23. Y. H. Hwang and P. J. Lee. Public key encryption with conjunctive keyword search and its extension to a multi-user system. In T. Takagi, editor, *Pairing-Based Cryptography – Pairing 2007*, volume 4575 of *LNCS*, pages 2–22. Springer, 2007.

24. V. Iovino and G. Persiano. Hidden-vector encryption with groups of prime order. In S. D. Galbraith and K. G. Paterson, editors, *Pairing-Based Cryptography – Pairing 2008*, volume 5209 of *LNCS*, pages 75–88. Springer, 2008.
25. J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In N. P. Smart, editor, *Advances in cryptology — EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 146–162. Springer, 2008.
26. A. B. Lewko and B. Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In D. Micciancio, editor, *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010*, volume 5978 of *Lecture Notes in Computer Science*, pages 455–479. Springer, 2010.
27. J. M. G. Nieto, M. Manulis, and D. Sun. Forward-secure hierarchical predicate encryption. In *Pairing-Based Cryptography – Pairing 2012*, volume 7708 of *LNCS*, pages 83–101. Springer, 2012.
28. M. Nishioka. Perfect keyword privacy in PEKS systems. In T. Takagi, G. Wang, Z. Qin, S. Jiang, and Y. Yu, editors, *Provable Security - 6th International Conference, ProvSec 2012*, volume 7496 of *Lecture Notes in Computer Science*, pages 175–192. Springer, 2012.
29. A. Raghunathan, G. Segev, and S. P. Vadhan. Deterministic public-key encryption for adaptively chosen plaintext distributions. In T. Johansson and P. Q. Nguyen, editors, *Advances in Cryptology — EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 93–110. Springer, 2013.
30. H. S. Rhee, J. H. Park, W. Susilo, and D. H. Lee. Trapdoor security in a searchable public-key encryption scheme with a designated tester. *J. Syst. Softw.*, 83(5):763–771, 2010.
31. E. Shi, J. Bethencourt, T-H. H. Chan, D. Song, and A. Perrig. Multi-dimensional range query over encrypted data. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, pages 350–364. IEEE Computer Society, 2007.
32. D. Yao, N. Fazio, Y. Dodis, and A. Lysyanskaya. Id-based encryption for complex hierarchies with applications to forward security and broadcast encryption. In V. Atluri, B. Pfitzmann, and P. D. McDaniel, editors, *Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS 2004*, pages 354–363. ACM, 2004.

Appendix I:

Proof of Lemma 1. Let Y_0, Y_1 be defined as follows.

$$Y_0 = (\Gamma^*,\ g_p^{b_1} \cdot g_q^{s_2},\ g_p^{a_1 b_1} \cdot g_q^{s_3}),\ \ Y_1 = (\Gamma^*,\ g_p^{b_1} \cdot g_q^{s_2},\ g_p^{r} \cdot g_q^{s_3})$$

It is clear that, given Y_b, the attacker's advantage in telling b is not larger than ϵ. Based on the standard hybrid argument, it is clear the lemma holds for the first scenario.

For the second scenario, we carry out the proof by an induction on L. Referring to the X_0, X_1 in Fig. 3, let $\alpha_0 = \Pr[\mathcal{A}(\Gamma^*, X_0) = 1]$ and $\alpha_1 = \Pr[\mathcal{A}(\Gamma^*, X_1) = 1]$. It is clear that $|\alpha_0 - \alpha_1| = 2\epsilon$. First, we consider the case when $L = 2$.

$$\alpha_1^{(2)} = \Pr[\mathcal{A}(\Gamma^*,\ g_p^{b_1} \cdot g_q^{s_1},\ g_p^{a_1 b_1} \cdot g_q^{t_1},\ g_p^{b_2} \cdot g_q^{s_2}, g_p^{a_1 b_2} \cdot g_q^{t_2}) = 1]$$

$$\alpha_2^{(2)} = \Pr[\mathcal{A}(\Gamma^*,\ g_p^{b_1} \cdot g_q^{s_1},\ g_p^{r_1} \cdot g_q^{t_1}, g_p^{b_2} \cdot g_q^{s_2},\ g_p^{r_2} \cdot g_q^{t_2}) = 1]$$

Let $\beta^{(2)}$ be defined as follows.

$$\beta^{(2)} = \Pr[\mathcal{A}(\Gamma^*,\ g_p^{b_1} \cdot g_q^{s_1},\ g_p^{a_1 b_1} \cdot g_q^{t_1},\ g_p^{b_2} \cdot g_q^{s_2},\ g_p^{r} \cdot g_q^{t_2}) = 1]$$

It is straightforward to verify that $|\alpha_1^{(2)} - \beta^{(2)}| \leq 2\epsilon$ and $|\alpha_2^{(2)} - \beta^{(2)}| \leq 2\epsilon$ based on the Composite-DDH-II assumption. Therefore, we have $|\alpha_1^{(2)} - \alpha_2^{(2)}| \leq 4 \cdot \epsilon$.

Suppose that $|\alpha_1^{(n)} - \alpha_2^{(n)}| \leq 2n \cdot \epsilon$, we prove that $|\alpha_1^{(n+1)} - \alpha_2^{(n+1)}| \leq 2(n+1) \cdot \epsilon$.

$$\alpha_1^{(n+1)} = \Pr[\mathcal{A}(\Gamma^*, g_p^{b_1} \cdot g_q^{s_1}, g_p^{a_1 b_1} \cdot g_q^{t_1}, \cdots, g_p^{b_n} \cdot g_q^{s_n}, g_p^{a_1 b_n} \cdot g_q^{t_n}, g_p^{b_{n+1}} \cdot g_q^{s_{n+1}}, g_p^{a_1 b_{n+1}} \cdot g_q^{t_{n+1}}) = 1]$$

$$\alpha_2^{(n+1)} = \Pr[\mathcal{A}(\Gamma^*, g_p^{b_1} \cdot g_q^{s_1}, g_p^{r_1} \cdot g_q^{t_1}, \cdots, g_p^{b_n} \cdot g_q^{s_n}, g_p^{r_n} \cdot g_q^{t_n}, g_p^{b_{n+1}} \cdot g_q^{s_{n+1}}, g_p^{r_{n+1}} \cdot g_q^{t_{n+1}}) = 1]$$

Let $\beta^{(n+1)}$ be defined as follows.

$$\beta^{(n+1)} = \Pr[\mathcal{A}(\Gamma^*, g_p^{b_1} \cdot g_q^{s_1}, g_p^{r_1} \cdot g_q^{t_1}, \cdots, g_p^{b_n} \cdot g_q^{s_n}, g_p^{r_n} \cdot g_q^{t_n}, g_p^{b_n \cdot r} \cdot g_q^{s_{n+1}}, g_p^{r_n \cdot r} \cdot g_q^{t_{n+1}}) = 1]$$

It is straightforward to verify that $|\alpha_2^{(n+1)} - \beta^{(n+1)}| \leq 2\epsilon$ based on the Composite-DDH-II assumption. With respect to $\alpha_1^{(n+1)}$ and $\beta^{(n+1)}$, we observe that the last the last two terms $(g_p^{b_{n+1}} \cdot g_q^{s_{n+1}}, g_p^{a_1 b_{n+1}} \cdot g_q^{t_{n+1}})$ and $(g_p^{b_n \cdot r} \cdot g_q^{s_{n+1}}, g_p^{r_n \cdot r} \cdot g_q^{t_{n+1}})$ can be unanimously generated by their previous terms. Therefore, we have $|\alpha_1^{(n+1)} - \beta^{(n+1)}| = |\alpha_1^{(n)} - \alpha_2^{(n)}|$ based on the fact that $\alpha_2^{(n)}$ is identical to $\beta^{(n+1)}$ without $(g_p^{b_n \cdot r} \cdot g_q^{s_{n+1}}, g_p^{r_n \cdot r} \cdot g_q^{t_{n+1}})$. Now, we have $|\alpha_1^{(n+1)} - \alpha_2^{(n+1)}| \leq 2\epsilon + |\alpha_1^{(n)} - \alpha_2^{(n)}| \leq 2(n+1) \cdot \epsilon$.

To sum up, we have $|\alpha_1^{(L)} - \alpha_2^{(L)}| \leq 2L \cdot \epsilon$, so that $|\Pr[b' = b] - \frac{1}{2}| \leq L \cdot \epsilon$ in the attack game defined in Fig. 4. The lemma now follows. □

Appendix II:

Proof of Theorem 1. Referring to the attack game for forward-secure trapdoor unlinkability, as shown in Fig. 5, the attacker is given the trapdoor vector $\boldsymbol{T}_b$ to guess b. The TrapGen oracle is not helpful to the attacker, since it is trivial to rerandomize any give trapdoor to obtain a new trapdoor for the same keyword. The received challenges by the attacker are in the following form. Let $\boldsymbol{x}_0 = (\boldsymbol{x}_0^{(1)}, \boldsymbol{x}_0^{(2)}, \cdots, \boldsymbol{x}_0^{(L)}) \overset{\mathbb{D}_0}{\leftarrow} \mathcal{W}^L$ and $\boldsymbol{x}_1 = (\boldsymbol{x}_1^{(1)}, \boldsymbol{x}_1^{(2)}, \cdots, \boldsymbol{x}_1^{(L)}) \overset{\mathbb{D}_1}{\leftarrow} \mathcal{W}^L$.

$$\boxed{\text{b=0}}: \quad ((g_p^{r_1^{(1)}} \cdot g_w'^{(1)}, \mathsf{H}(\boldsymbol{x}_0^{(1)})^{r_1^{(1)}} \cdot g_w''^{(1)}), \cdots, (g_p^{r_1^{(L)}} \cdot g_w'^{(L)}, \mathsf{H}(\boldsymbol{x}_0^{(L)})^{r_1^{(L)}} \cdot g_w''^{(L)}))$$

$$\boxed{\text{b=1}}: \quad ((g_p^{r_1^{(1)}} \cdot g_w'^{(1)}, \mathsf{H}(\boldsymbol{x}_1^{(1)})^{r_1^{(1)}} \cdot g_w''^{(1)}), \cdots, (g_p^{r_1^{(L)}} \cdot g_w'^{(L)}, \mathsf{H}(\boldsymbol{x}_1^{(L)})^{r_1^{(L)}} \cdot g_w''^{(L)}))$$

The rest of the proof is identical to the proof of Theorem 2, where we show that $|\Pr[b' = b] - \frac{1}{2}|$ is negligible based on the Weak Composite-DDH assumption in the random oracle model. The theorem follows. □

Appendix III:

Proof of Theorem 2. Recall that in the attack game, shown in Fig. 10, the attacker is given *Msk* in Step 4. In our proof, we simply give *Msk* to the attacker in Step 2, so that there is no need for the attacker to submit any oracle queries (except for the random oracle H) to the challenger anymore. Moreover, for any secret key sk_{id} as described in Fig.18, if we give something like $(x \cdot g_p^r, y \cdot \mathsf{H}(id)^r)$ for $x, y \xleftarrow{\$} \mathbb{G}_q$ and $r \xleftarrow{\$} \mathbb{Z}_n$ directly to the attacker then the latter can extend it to the full form. After the simplification, the received challenges by the attacker are in the following form.

$$\boxed{\text{b=0}}: \ ((x_1 \cdot g_p^{r_1},\ y_1 \cdot \mathsf{H}(\boldsymbol{id}_0^{(1)})^{r_1}),\ \cdots,\ (x_L \cdot g_p^{r_L},\ y_L \cdot \mathsf{H}(\boldsymbol{id}_0^{(L)})^{r_L}))$$

$$\boxed{\text{b=1}}: \ ((x_1 \cdot g_p^{r_1},\ y_1 \cdot \mathsf{H}(\boldsymbol{id}_1^{(1)})^{r_1}),\ \cdots,\ (x_L \cdot g_p^{r_L},\ y_L \cdot \mathsf{H}(\boldsymbol{id}_1^{(L)})^{r_L}))$$

Next, we carry out the proof by an induction on L.

Case $L = 1$. In the faithful game, the challenge is $sk_{\boldsymbol{id}_b} = sk_{\boldsymbol{id}_b^{(1)}}$, where

$$\boldsymbol{id}_0 = \boldsymbol{id}_0^{(1)} \xleftarrow{\mathbb{D}_0} \mathcal{I},\ sk_{\boldsymbol{id}_0^{(1)}} = (x_1 \cdot g_p^{r_1},\ y_1 \cdot \mathsf{H}(\boldsymbol{id}_0^{(1)})^{r_1})$$

$$\boldsymbol{id}_1 = \boldsymbol{id}_1^{(1)} \xleftarrow{\mathbb{D}_1} \mathcal{I},\ sk_{\boldsymbol{id}_1^{(1)}} = (x_1 \cdot g_p^{r_1},\ y_1 \cdot \mathsf{H}(\boldsymbol{id}_1^{(1)})^{r_1})$$

Let the attacker's advantage be ϵ_1. Consider a new game, where the challenge $sk_{\boldsymbol{id}_b}$ is generated as follows.

$$\alpha \xleftarrow{\$} \mathbb{G}_p,\ sk_{\boldsymbol{id}_0^{(1)}} = (x_1 \cdot g_p^{r_1},\ y_1 \cdot \alpha^{r_1})$$

$$\alpha \xleftarrow{\$} \mathbb{G}_p,\ sk_{\boldsymbol{id}_1^{(1)}} = (x_1 \cdot g_p^{r_1},\ y_1 \cdot \alpha^{r_1})$$

Suppose the attacker issues h queries to the random oracle H in the game. The new game is identical to the original one with the probability $1 - \frac{h}{2^\lambda}$, where $\frac{h}{2^\lambda}$ is the probability that the attacker has queried one of the identities in $\boldsymbol{id}_0$ and $\boldsymbol{id}_1$ to the random oracle. It is clear that the attacker's advantage is 0 when the games are identical. As a result, $\epsilon_1 \leq \frac{h}{2^\lambda}$.

Case $L = 2$. In the faithful game, the challenge is $sk_{\boldsymbol{id}_b} = (sk_{\boldsymbol{id}_b^{(1)}}, sk_{\boldsymbol{id}_b^{(2)}})$, where

$$\boldsymbol{id}_0 = (\boldsymbol{id}_0^{(1)}, \boldsymbol{id}_0^{(2)}) \xleftarrow{\mathbb{D}_0} (\mathcal{I}, \mathcal{I}), sk_{\boldsymbol{id}_0^{(1)}} = (x_1 \cdot g_p^{r_1}, y_1 \cdot \mathsf{H}(\boldsymbol{id}_0^{(1)})^{r_1}), sk_{\boldsymbol{id}_0^{(2)}} = (x_2 \cdot g_p^{r_2}, y_2 \cdot \mathsf{H}(\boldsymbol{id}_0^{(2)})^{r_2})$$

$$\boldsymbol{id}_1 = (\boldsymbol{id}_1^{(1)}, \boldsymbol{id}_1^{(2)}) \xleftarrow{\mathbb{D}_1} (\mathcal{I}, \mathcal{I}), sk_{\boldsymbol{id}_1^{(1)}} = (x_1 \cdot g_p^{r_1}, y_1 \cdot \mathsf{H}(\boldsymbol{id}_1^{(1)})^{r_1}), sk_{\boldsymbol{id}_1^{(2)}} = (x_2 \cdot g_p^{r_2}, y_2 \cdot \mathsf{H}(\boldsymbol{id}_1^{(2)})^{r_2})$$

Let the attacker's advantage be ϵ_2. Consider a new game, which is faithful except for the challenge generation.

- If $\boldsymbol{id}_0^{(1)} \neq \boldsymbol{id}_0^{(2)}$, the challenge $sk_{\boldsymbol{id}_0}$ is generated as follows.

$$\alpha, \beta \xleftarrow{\$} \mathbb{G}_p,\ sk_{\boldsymbol{id}_0^{(1)}} = (x_1 \cdot g_p^{r_1},\ y_1 \cdot \alpha^{r_1}),\ sk_{\boldsymbol{id}_0^{(2)}} = (x_2 \cdot g_p^{r_2},\ y_2 \cdot \beta^{r_2})$$

Otherwise, the challenge $sk_{\boldsymbol{id}_0}$ is generated as follows.

$$\alpha \xleftarrow{\$} \mathbb{G}_p,\ sk_{\boldsymbol{id}_0^{(1)}} = (x_1 \cdot g_p^{r_1},\ y_1 \cdot \alpha^{r_1}),\ sk_{\boldsymbol{id}_0^{(2)}} = (x_2 \cdot g_p^{r_2},\ y_2 \cdot \alpha^{r_2})$$

- The challenge $sk_{\boldsymbol{id}_1}$ is always generated as follows.

$$\alpha, \beta \xleftarrow{\$} \mathbb{G}_p,\ \ sk_{\boldsymbol{id}_1^{(1)}} = (x_1 \cdot g_p^{r_1},\ \ y_1 \cdot \alpha^{r_1}),\ \ sk_{\boldsymbol{id}_1^{(2)}} = (x_2 \cdot g_p^{r_2},\ \ y_2 \cdot \beta^{r_2})$$

Suppose the attacker issues h queries to the random oracle H in the game. The new game is identical to the original one with the probability $1 - \frac{2h}{2^\lambda}$, where $\frac{2h}{2^\lambda}$ is the probability that the attacker has queried one of the identities in $\boldsymbol{id}_0$ and $\boldsymbol{id}_1$ to the random oracle. When the games are identical, we can compute the attacker's advantage by considering two cases.

- One case is $\boldsymbol{id}_0^{(1)} \neq \boldsymbol{id}_0^{(2)}$. Let $p_1 = \Pr[\boldsymbol{id}_0^{(1)} \neq \boldsymbol{id}_0^{(2)}]$ according to $\mathbb{D}_0$. In this case, the attacker's advantage is 0.
- The other case is $\boldsymbol{id}_0^{(1)} = \boldsymbol{id}_0^{(2)}$. Let $p_2 = \Pr[\boldsymbol{id}_0^{(1)} = \boldsymbol{id}_0^{(2)}]$ according to $\mathbb{D}_0$. In this case, the attacker's advantage is exactly Adv_{wcddh}, which is the attacker's advantage in the Weak Composite-DDH assumption.

Combining the two cases, the attacker's overall advantage is $p_2 \cdot Adv_{wcddh}$ when the new game is identical to the original one. As a result, $\epsilon_2 \leq \frac{2h}{2^\lambda} + Adv_{wcddh}$.

Case $L = 3$. In the faithful game, the challenge is $sk_{\boldsymbol{id}_b} = (sk_{\boldsymbol{id}_b^{(1)}}, sk_{\boldsymbol{id}_b^{(2)}}, sk_{\boldsymbol{id}_b^{(3)}})$, where

$$\boldsymbol{id}_0 = (\boldsymbol{id}_0^{(1)}, \boldsymbol{id}_0^{(2)}, \boldsymbol{id}_0^{(3)}) \xleftarrow{\mathbb{D}_0} (\mathcal{I}, \mathcal{I}, \mathcal{I}),\ \ sk_{\boldsymbol{id}_0^{(1)}} = (x_1 \cdot g_p^{r_1},\ \ y_1 \cdot \mathsf{H}(\boldsymbol{id}_0^{(1)})^{r_1})$$

$$sk_{\boldsymbol{id}_0^{(2)}} = (x_2 \cdot g_p^{r_2},\ \ y_2 \cdot \mathsf{H}(\boldsymbol{id}_0^{(2)})^{r_2}),\ \ sk_{\boldsymbol{id}_0^{(3)}} = (x_3 \cdot g_p^{r_3},\ \ y_3 \cdot \mathsf{H}(\boldsymbol{id}_0^{(3)})^{r_3})$$

$$\boldsymbol{id}_1 = (\boldsymbol{id}_1^{(1)}, \boldsymbol{id}_1^{(2)}, \boldsymbol{id}_1^{(3)}) \xleftarrow{\mathbb{D}_1} (\mathcal{I}, \mathcal{I}, \mathcal{I}),\ \ sk_{\boldsymbol{id}_1^{(1)}} = (x_1 \cdot g_p^{r_1},\ \ y_1 \cdot \mathsf{H}(\boldsymbol{id}_1^{(1)})^{r_1})$$

$$sk_{\boldsymbol{id}_1^{(2)}} = (x_2 \cdot g_p^{r_2},\ \ y_2 \cdot \mathsf{H}(\boldsymbol{id}_1^{(2)})^{r_2}),\ \ sk_{\boldsymbol{id}_1^{(3)}} = (x_3 \cdot g_p^{r_3},\ \ y_3 \cdot \mathsf{H}(\boldsymbol{id}_1^{(3)})^{r_3})$$

Let the attacker's advantage be ϵ_3. Consider a new game, which is faithful except for the challenge generation.

- For $sk_{\boldsymbol{id}_0}$, sample $\alpha_1 \xleftarrow{\$} \mathbb{G}_p$. If $\boldsymbol{id}_0^{(1)} = \boldsymbol{id}_0^{(2)}$, set $\alpha_2 = \alpha_1$, otherwise sample $\alpha_2 \xleftarrow{\$} \mathbb{G}_p$. If $\boldsymbol{id}_0^{(3)} = \boldsymbol{id}_0^{(i)}$ for some $i \in \{1, 2\}$, set $\alpha_3 = \alpha_i$, otherwise sample $\alpha_3 \xleftarrow{\$} \mathbb{G}_p$. $sk_{\boldsymbol{id}_0}$ is computed as follows.

$$sk_{\boldsymbol{id}_1^{(1)}} = (x_1 \cdot g_p^{r_1}, y_1 \cdot \alpha_1^{r_1}), sk_{\boldsymbol{id}_1^{(2)}} = (x_2 \cdot g_p^{r_2}, y_2 \cdot \alpha_2^{r_2}), sk_{\boldsymbol{id}_1^{(3)}} = (x_3 \cdot g_p^{r_3}, y_3 \cdot \alpha_3^{r_3}) \quad (1)$$

- The challenge $sk_{\boldsymbol{id}_1}$is generated as follows.

$$\alpha, \beta, \gamma \xleftarrow{\$} \mathbb{G}_p,$$

$$sk_{\boldsymbol{id}_1^{(1)}} = (x_1 \cdot g_p^{r_1}, y_1 \cdot \alpha^{r_1}), sk_{\boldsymbol{id}_1^{(2)}} = (x_2 \cdot g_p^{r_2}, y_2 \cdot \beta^{r_2}), sk_{\boldsymbol{id}_1^{(3)}} = (x_3 \cdot g_p^{r_3}, y_3 \cdot \gamma^{r_3}) \quad (2)$$

Suppose the attacker issues h queries to the random oracle H in the game. It is clear that this new game is identical to the original one with the probability $1-\frac{3h}{2^\lambda}$, where $\frac{3h}{2^\lambda}$ is the probability that the attacker has queried one of the identities in $\boldsymbol{id}_0$ and $\boldsymbol{id}_1$ to the random oracle. When the games are identical, we can compute the attacker's advantage by considering two cases.

- One case is $\boldsymbol{id}_0^{(1)} \neq \boldsymbol{id}_0^{(2)} \neq \boldsymbol{id}_0^{(3)}$. Let p_1 be the probability of this case according to $\mathbb{D}_0$. In this case, the attacker's advantage is 0.
- The other case is $\boldsymbol{id}_0^{(i)} = \boldsymbol{id}_0^{(j)}$ for some $1 \leq i \neq j \leq 3$. Let p_2 be the probability of this case according to $\mathbb{D}_0$. Let $sk_{\boldsymbol{id}_0^{(z)}}$ be the left element in $sk_{\boldsymbol{id}_0}$. Next, we need to compute the probability q_1, which is the probability that the attacker outputs 0 given $\{sk_{\boldsymbol{id}_0^{(i)}}, sk_{\boldsymbol{id}_0^{(j)}}, sk_{\boldsymbol{id}_0^{(z)}}\}$ in the form of Equation (1).
    - Let q_2 be the probability that the attacker outputs 0 given $\{sk^*_{\boldsymbol{id}_0^{(i)}}, sk_{\boldsymbol{id}_0^{(j)}}, sk_{\boldsymbol{id}_0^{(z)}}\}$, where $sk^*_{\boldsymbol{id}_0^{(i)}}$ generated by by replacing the α_i with $\beta \xleftarrow{\$} \mathbb{G}_p$ in the generation of $sk_{\boldsymbol{id}_0^{(i)}}$. we have $|\frac{q_1+1-q_2}{2} - \frac{1}{2}| \leq Adv_{wcddh}$ for two reasons: (1) $|\frac{q_1+1-q_2}{2} - \frac{1}{2}|$ is the attacker's advantage in distinguishing these two key vectors; (2) Given either $\{sk_{\boldsymbol{id}_0^{(i)}}, sk_{\boldsymbol{id}_0^{(j)}}\}$ or $\{sk^*_{\boldsymbol{id}_0^{(i)}}, sk_{\boldsymbol{id}_0^{(j)}}\}$ the attacker can simulate $sk_{\boldsymbol{id}_0^{(z)}}$. This means the attacker's advantage is Adv_{wcddh}.
    - Let q_3 be the probability that the attacker outputs 0 given $sk_{\boldsymbol{id}_1}$ in the form of Equation (2). We have $|\frac{q_2+1-q_3}{2} - \frac{1}{2}| \leq |\epsilon_2 - \frac{2h}{2^\lambda}| \leq Adv_{wcddh}$.

  In summary, the attacker's advantage in this case is $|\frac{q_1+1-q_3}{2} - \frac{1}{2}| \leq 2Adv_{wcddh}$.

Combining both cases, the attacker's advantage is $2Adv_{wcddh}$ when the new game is identical to the original one. As a result, the attacker's advantage in the original game is

$$\epsilon_3 \leq \frac{3h}{2^\lambda} + 2Adv_{wcddh}. \tag{3}$$

<u>Reduction from $L = n$ to $L = n + 1$</u>. Suppose when $L = n$, the attacker has the advantage ϵ_n. Next, we compute the attacker's advantage ϵ_{n+1} when $L = n + 1$. Based on the faithful game, consider a new game, where the hash values of identities in the challenge are replaced with randomly chosen elements from $\mathbb{G}$ in the same manner as in the case of $L = 3$ (basically, if two identities for $\boldsymbol{id}_0$ are the same then they use the same random value). This will make the new game be identical with the original one with the probability $1 - \frac{(n+1)\cdot h}{2^\lambda}$, where $\frac{(n+1)\cdot h}{2^\lambda}$ is the probability that the attacker has queried one of the identities in $\boldsymbol{id}_0$ and $\boldsymbol{id}_1$ to the random oracle. Next, we can compute the attacker's advantage by considering two cases.

- One case is $\boldsymbol{id}_0^{(1)} \neq \boldsymbol{id}_0^{(2)} \neq \cdots \neq \boldsymbol{id}_0^{(n+1)}$. Let p_1 be the probability of this case according to $\mathbb{D}_0$. In this case, the attacker's advantage is 0.

- The other case is $\boldsymbol{id}_0^{(i)} = \boldsymbol{id}_0^{(j)}$ for some $1 \leq i \neq j \leq n+1$. Let p_2 be the probability of this case according to $\mathbb{D}_0$. In this case, the attacker's advantage is $|\epsilon_n - \frac{nh}{2^\lambda}| + Adv_{wcddh}$ for the same reason as in computing q_1 in the case of $L = 3$.

Combining both cases, the attacker's advantage is $p_2(|\epsilon_n - \frac{nh}{2^\lambda}| + Adv_{wcddh}) \leq |\epsilon_n - \frac{nh}{2^\lambda}| + Adv_{wcddh}$ when the new game is identical to the original one. As a result, the attacker's advantage in the original game is

$$\epsilon_{n+1} \leq |\epsilon_n - \frac{nh}{2^\lambda}| + Adv_{wcddh} + \frac{(n+1)\cdot h}{2^\lambda}. \tag{4}$$

Conclusion. Based on the inequalities (3) and (4), we have $\epsilon_L \leq +(L-1)\cdot Adv_{wcddh} + \frac{L\cdot h}{2^\lambda}$. The theorem is proven. □

Appendix IV:

Proof of Theorem 3. For simplicity, suppose *Msk* is public. Moreover, for any secret key sk_{id} as described in Fig.20, if we give something like $(x \cdot g_p^r, y \cdot (g_0 g_1^{\mathsf{H}(id)})^r)$ for $x, y \xleftarrow{\$} \mathbb{G}_q$ and $r \xleftarrow{\$} \mathbb{Z}_n$ directly to the attacker then the latter can extend it to the full form. With this simplification, referring to the attack game in Fig. 10, the challenge is in the following form, where $x_1, \cdots, x_L, y_1, \cdots, y_L \xleftarrow{\$} \mathbb{G}_q$ and $s_1, \cdots, s_L, r_1, \cdots, r_L \xleftarrow{\$} \mathbb{Z}_n$.

$$\boxed{\text{b=0}}: \quad (x_1 \cdot g_p^{r_1}, y_1 \cdot (g_0 g_1^{\mathsf{H}(\boldsymbol{id}_0^{(1)})})^{r_1}; \cdots; x_L \cdot g_p^{r_L}, y_L \cdot (g_0 g_1^{\mathsf{H}(\boldsymbol{id}_0^{(L)})})^{r_L})$$

$$\boxed{\text{b=1}}: \quad (x_1 \cdot g_p^{r_1}, y_1 \cdot (g_0 g_1^{\mathsf{H}(\boldsymbol{id}_1^{(1)})})^{r_1}; \cdots; x_L \cdot g_p^{r_L}, y_L \cdot (g_0 g_1^{\mathsf{H}(\boldsymbol{id}_1^{(L)})})^{r_L})$$

Since H is collision-resistant so that it does not change the min-entropy of the input identities. As a result, we can conclude that $g_0 g_1^{\mathsf{H}(\boldsymbol{id}_0^{(i)})}$ for every $1 \leq i \leq L$ has min-entropy λ. On the other hand, $g_0 g_1^{\mathsf{H}(\boldsymbol{id}_1^{(i)})}$ for every $1 \leq i \leq L$ is uniformly distributed. Based on these facts, $|\Pr[b' = b] - \frac{1}{2}|$ is negligible if the correlated Composite-DDH-II assumption holds. The theorem follows. □