Trapdoor Privacy in Asymmetric Searchable Encryption Schemes

Afonso Arriaga, Qiang Tang, and Peter Ryan

SnT, University of Luxembourg
{afonso.delerue,qiang.tang,peter.ryan}@uni.lu

Abstract. Asymmetric searchable encryption allows searches to be carried over ciphertexts, through delegation, and by means of trapdoors issued by the owner of the data. Public Key Encryption with Keyword Search (PEKS) is a primitive with such functionality that provides delegation of exact-match searches. As it is important that ciphertexts preserve data privacy, it is also important that trapdoors do not expose the user's search criteria. The difficulty of formalizing a security model for trapdoor privacy lies in the verification functionality, which gives the adversary the power of verifying if a trapdoor encodes a particular keyword. In this paper, we provide a broader view on what can be achieved regarding trapdoor privacy in asymmetric searchable encryption schemes, and bridge the gap between previous definitions, which give limited privacy guarantees in practice against search patterns. Since it is well-known that PEKS schemes can be trivially constructed from any Anonymous IBE scheme, we propose the security notion of Key Unlinkability for IBE, which leads to strong guarantees of trapdoor privacy in PEKS, and we construct a scheme that achieves this security notion.

Keywords: Asymmetric Searchable Encryption, PEKS, Trapdoor Privacy, Function Privacy, Search Pattern Privacy, Key Unlinkability.

1 Introduction

As cloud services become increasingly popular, security concerns arise from exposing the user's data to third-party service providers. Encryption can be used to protect the user's data privacy, but usability is sacrificed if not even the most basic operations, such as searching over the user's data, can be delegated to the service provider. In the public key setting, Boneh et al. [6] were the first to propose a primitive to tackle this problem. They called it Public Key Encryption with Keyword Search (PEKS), a primitive that provides delegation of *exact-match* searches over ciphertexts. A typical scenario where this primitive can bring great benefits to users (and consequently to service providers wishing to increase their customer base as well) is that of any email system.

Suppose user Alice stores her emails in the servers of some email service provider, so that she can access them from either her laptop or her smartphone. Alice does not trust the service provider or fears that government agencies may

D. Pointcheval and D. Vergnaud (Eds.): AFRICACRYPT 2014, LNCS 8469, pp. 31–50, 2014.

require the service provider to hand over all her data. Using standard public key encryption, any user with Alice's public key can send her encrypted emails that only she can decrypt. For Alice to find a particular email later on, the sender could also attach to the email some *searchable ciphertexts*, produced from a PEKS scheme, with keywords that Alice might use when searching for this email. These ciphertexts are searchable upon delegation, meaning that only Alice can authorize the email service provider to search on her behalf by issuing a trapdoor that encodes Alice's search criteria (e.g. ciphertexts that encrypt the keyword "project xx123 - meeting"), generated from her own secret key. The service provider searches through all Alice's emails for those containing searchable ciphertexts that match the issued trapdoor, and returns to her only those with a positive match.

Many efforts have been put in asymmetric searchable encryption in general, as surveyed in [16], most towards more efficient PEKS schemes (or relying on weaker assumptions) or towards primitives with more flexible search queries, such as conjunctive, disjunctive, subset and inner product types of queries. Until recently [14,9,10], the concern was always to preserve data privacy in the ciphertexts and no attention was paid to possible information leakage from the trapdoors. In fact, some schemes, as the statistically consistent scheme proposed in [1], include the keyword itself in the trapdoor. In this paper we focus on defining trapdoor privacy for PEKS and constructing a scheme that provably stands up to the definition. Nevertheless, the definition can easily be extended to asymmetric searchable encryption in general [13].

The difficulty of formalizing a security model for trapdoor privacy lies in the verification functionality of PEKS, which in the public key setting depends on the trapdoor itself and ciphertexts created from publicly known parameters. This provides to any adversary the power to verify if a trapdoor encodes a particular keyword. (The adversary encrypts the chosen keyword under the public key associated with the trapdoor; if the ciphertext matches the trapdoor then the trapdoor encodes the chosen keyword.) Therefore, an offline dictionary attack can always be launched, putting aside the possibility of formalizing the security notion of trapdoor privacy as a traditional choose-then-guess indistinguishability game in the public-key setting - although possible in the symmetric setting [15]. In many cases, the keywords encoded in trapdoors are *sufficiently unpredictable* for a dictionary attack to be infeasible. So, defining the right notion of trapdoor privacy is crucial to guarantee that the user's privacy is fully protected.

Related Work. Abdalla et al. [1], by extending the results left implicit in [6], proposed a general black-box transformation from *Anonymous* Identity-Based Encryption (IBE) to PEKS, where the resulting PEKS scheme is secure in the traditional ciphertext indistinguishability sense. Identities and their secret keys in the original IBE scheme map to keywords and trapdoors in resulting PEKS scheme, respectively. The anonymity requirement informally states that ciphertexts leak no information regarding the identity of the recipient, leading to the commonly desired keyword-privacy guarantees over ciphertexts in PEKS. The standard notion of ciphertext indistinguishability in IBE leads to *Computational*

Consistency in the resulting PEKS scheme, which informally means that it is hard for computationally bounded adversaries to find two distinct keywords such that the trapdoors for the first keyword positively match the ciphertexts of the second keyword. (Note that if keywords are hashed before used in the scheme, *inconsistency* happens at least every time $\mathsf{H}(\mathsf{w}_1) = \mathsf{H}(\mathsf{w}_2)$, where $\mathsf{w}_1 \neq \mathsf{w}_2$.) We refer the reader to Section 2 for precise details on this transformation and to [1] for formal proofs.

This (black-box) transformation allows us to define a "dual" security notion for IBE that will lead to the desired trapdoor privacy notion in PEKS, and motivates the construction of IBE schemes that can provably satisfy it. This approach was also followed by [14,9,10], which, to the best of our knowledge, are the only works to address the concerns on trapdoor privacy in asymmetric searchable encryption.

Two distinct scenarios have to be considered to model trapdoor privacy. One in the presence of ciphertexts that positively match the trapdoors, and the other in the absence of such ciphertexts. Consider a toy example where the service provider possesses one ciphertext that belongs to Alice and two trapdoors that Alice issued for searches to be performed on her behalf. The service provider executes the test-search and one of the following cases occurs:

(a) Both trapdoors positively match the stored ciphertext, in which case the trapdoors encode the same keyword.
(b) Only one of the trapdoors match the ciphertext, in which case the trapdoors encode different keywords.
(c) None of the trapdoors positively match the stored ciphertext.

From cases (a) and (b), we can see that, in the presence of ciphertexts that match the trapdoors, an equality relation between the keywords encoded under the trapdoors can be determined trivially. In such cases, the notion of trapdoor privacy focus on revealing as little information as possible on the keywords themselves[1]. Recently, Boneh et al. [9] put forward two formal definitions of different strengths for IBE, inspired by the security definition given for Deterministic Encryption in [3]: *Function Privacy* and *Enhanced Function Privacy*. The latter leads to a security notion in PEKS (after the black-box transformation in [1]), which addresses this scenario.

Case (c) covers the scenario where trapdoors do not match any ciphertext. It is in Alice's best interest to hide her search pattern from the service provider. If the search pattern is revealed, the attacker could concentrate its resources on breaking the privacy of trapdoors encoding the most frequent keywords, which a priori are the most relevant to Alice. This issue is particularly important for PEKS due to the possibility of launching dictionary attacks. Nishioka [14] proposed a model denoted *Search Pattern Privacy*, which partially addresses this scenario. However, the model limits the distinguishing game to two trapdoors,

[1] Note that some information is inevitably leaked because of the verification functionality, e.g. if the trapdoor does not match a ciphertext which encrypts a particular known keyword, it means that the trapdoor does not encode this keyword.

which provides insufficient privacy guarantees in practice, considering that an actual attacker may have access to a much larger number of (possibly related) trapdoors. As we show in Section 4, and contrarily to intuition, the so-called hybrid argument does not apply here, unless trapdoors can be efficiently re-randomized. Also, after the transformation from IBE to PEKS, Function Privacy leads to a security definition that provides limited privacy guarantees against search patterns, since the resulting model prevents the adversary from being challenged with trapdoors encoding the same keyword.

Our Contributions. We formulate the "dual" notion of Search Pattern Privacy [14], which we call *Weak Key Unlinkability* for IBE. We then show that Weak Key Unlinkability is insufficient in practice. We do so by constructing a new Anonymous IBE scheme with Weak Key Unlinkability, based on the Anonymous IBE scheme by Boyen and Waters [12], and show that the resulting PEKS scheme (by applying black-box transformation in [1]) fails to hide search patterns when more than two trapdoors have been issued. We then propose a new security model, strictly stronger than Weak Key Unlinkability, which we call *Strong Key Unlinkability*. We compare the different notions of security and show that Key Unlinkability and Function Privacy [9] are orthogonal notions. Finally, we extend our IBE scheme to groups of composite order, and prove its security in the Strong Key Unlinkability model.

2 Preliminaries

Notation. We write $\mathsf{a} \leftarrow \mathsf{b}$ to denote the algorithmic action of assigning the value of b to the variable a. We use $\perp \notin \{0,1\}^\star$ to denote a special failure symbol. If $\mathbb{S}$ is a set, we write $\mathsf{a} \leftarrow_\$ \mathbb{S}$ for sampling a from $\mathbb{S}$ uniformly at random. If $\mathbb{X}$ is a joint probability distribution with L random variables, we write $(\mathsf{x}_1, ..., \mathsf{x}_\mathsf{L}) \leftarrow_\$ \mathbb{X}$ for sampling $(\mathsf{x}_1, ..., \mathsf{x}_\mathsf{L})$ from $\mathbb{X}$. If $\mathcal{A}$ is a probabilistic algorithm we write $\mathsf{a} \leftarrow_\$ \mathcal{A}(\mathsf{i}_1, \mathsf{i}_2, \ldots, \mathsf{i}_\mathsf{n})$ for the action of running $\mathcal{A}$ on inputs $\mathsf{i}_1, \mathsf{i}_2, \ldots, \mathsf{i}_\mathsf{n}$ with random coins, and assigning the result to a. If a is a variable, $|\mathsf{a}|$ denotes the length in bits of its representation. We denote by $\mathsf{a}||\mathsf{b}$ the concatenation of variables a and b, represented as bit-strings.

Games. In this paper we use the code-based game-playing language [4]. Each game has an Initialize and a Finalize procedure. It also has specifications of procedures to respond to an adversary's various queries. A game is run with an adversary $\mathcal{A}$ as follows. First Initialize runs and its outputs are passed to $\mathcal{A}$. Then $\mathcal{A}$ runs and its oracle queries are answered by the procedures of the game. When $\mathcal{A}$ terminates, its output is passed to Finalize, which returns the outcome of the game. In each game, we restrict attention to legitimate adversaries, which is defined specifically for each game. We use lists as data structures to keep relevant state in the games. The empty list is represented by empty square brackets [ ]. We denote by $\mathsf{list} \leftarrow \mathsf{a} : \mathsf{list}$ the action of appending element a to the head of list. To access the value stored in index i of list and assign it to a, we write $\mathsf{a} \leftarrow \mathsf{list}[\mathsf{i}]$. To denote the number of elements in list, we use $|\mathsf{list}|$. Unless stated otherwise, lists are initialized empty and variables are first assigned with $\perp$.

2.1 Bilinear Groups

We first revise pairings over prime-order groups and the associated *Decision Bilinear Diffie-Hellman* (DBDH) and *Decision Linear* (DLIN) assumptions [7,5]. We then revise pairings over composite-order groups [8], introduce the new *Composite Decision Diffie-Hellman* (CDDH) assumption, and show that this assumption is weaker than the well-established *Composite 3-party Diffie-Hellman* (C3DH) assumption made in [11].

Bilinear Groups of Prime Order

Definition 1. *A prime-order bilinear group generator is an algorithm $\mathcal{G}_{\mathcal{P}}$ that takes as input a security parameter λ and outputs a description $\Gamma = (\mathsf{p}, \mathbb{G}, \mathbb{G}_{\mathsf{T}}, \mathbf{e}, \mathsf{g})$ where $\mathbb{G}$ and $\mathbb{G}_{\mathsf{T}}$ are groups of order p with efficiently-computable group laws, where p is a λ-bit prime, g is a generator of $\mathbb{G}$ and $\mathbf{e}$ is an efficiently-computable bilinear pairing $\mathbf{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_{\mathsf{T}}$.*

Definition 2. *Let $\Gamma = (\mathsf{p}, \mathbb{G}, \mathbb{G}_{\mathsf{T}}, \mathbf{e}, \mathsf{g})$ be the description output by $\mathcal{G}_{\mathcal{P}}(\lambda)$. We say the DBDH assumption holds for description Γ if, for every PPT adversary $\mathcal{A}$, the following definition of advantage is negligible in λ.*

$$\mathbf{Adv}^{\mathsf{DBDH}}_{\Gamma,\mathcal{A}} := 2 \cdot \Pr[\mathsf{DBDH} \Rightarrow \mathsf{True}] - 1,$$

where game DBDH *is described in Fig. 1.*

procedure Initialize(λ):
$\Gamma \leftarrow_{\$} \mathcal{G}_{\mathcal{P}}(\lambda)$
$(\mathsf{p}, \mathbb{G}, \mathbb{G}_{\mathsf{T}}, \mathbf{e}, \mathsf{g}) \leftarrow \Gamma$
$\mathsf{z}_1 \leftarrow_{\$} \mathbb{Z}_{\mathsf{p}}$
$\mathsf{z}_2 \leftarrow_{\$} \mathbb{Z}_{\mathsf{p}}$
$\mathsf{z}_3 \leftarrow_{\$} \mathbb{Z}_{\mathsf{p}}$
$\mathsf{Z} \leftarrow_{\$} \mathbb{G}_{\mathsf{T}}$
$\mathsf{bit} \leftarrow_{\$} \{0,1\}$
if $\mathsf{bit} = 0$ return $(\Gamma, \mathsf{g}^{\mathsf{z}_1}, \mathsf{g}^{\mathsf{z}_2}, \mathsf{g}^{\mathsf{z}_3}, \mathbf{e}(\mathsf{g},\mathsf{g})^{\mathsf{z}_1\mathsf{z}_2\mathsf{z}_3})$
else return $(\Gamma, \mathsf{g}^{\mathsf{z}_1}, \mathsf{g}^{\mathsf{z}_2}, \mathsf{g}^{\mathsf{z}_3}, \mathsf{Z})$

procedure Finalize(bit'):
if $\mathsf{bit} = \mathsf{bit}'$ return True
else return False

Fig. 1. Game DBDH

procedure Initialize(λ):
$\Gamma \leftarrow_{\$} \mathcal{G}_{\mathcal{P}}(\lambda)$
$(\mathsf{p}, \mathbb{G}, \mathbb{G}_{\mathsf{T}}, \mathbf{e}, \mathsf{g}) \leftarrow \Gamma$
$\mathsf{z}_1 \leftarrow_{\$} \mathbb{Z}_{\mathsf{p}}$
$\mathsf{z}_2 \leftarrow_{\$} \mathbb{Z}_{\mathsf{p}}$
$\mathsf{z}_3 \leftarrow_{\$} \mathbb{Z}_{\mathsf{p}}$
$\mathsf{z}_4 \leftarrow_{\$} \mathbb{Z}_{\mathsf{p}}$
$\mathsf{Z} \leftarrow_{\$} \mathbb{G}_{\mathsf{T}}$
$\mathsf{bit} \leftarrow_{\$} \{0,1\}$
if $\mathsf{bit} = 0$ return $(\Gamma, \mathsf{g}^{\mathsf{z}_1}, \mathsf{g}^{\mathsf{z}_2}, \mathsf{g}^{\mathsf{z}_1\mathsf{z}_3}, \mathsf{g}^{\mathsf{z}_2\mathsf{z}_4}, \mathsf{g}^{\mathsf{z}_3+\mathsf{z}_4})$
else return $(\Gamma, \mathsf{g}^{\mathsf{z}_1}, \mathsf{g}^{\mathsf{z}_2}, \mathsf{g}^{\mathsf{z}_1\mathsf{z}_3}, \mathsf{g}^{\mathsf{z}_2\mathsf{z}_4}, \mathsf{Z})$

procedure Finalize(bit'):
if $\mathsf{bit} = \mathsf{bit}'$ return True
else return False

Fig. 2. Game DLIN

Definition 3. *Let $\Gamma = (\mathsf{p}, \mathbb{G}, \mathbb{G}_{\mathsf{T}}, \mathbf{e}, \mathsf{g})$ be the description output by $\mathcal{G}_{\mathcal{P}}(\lambda)$. We say the DLIN assumption holds for description Γ if, for every PPT adversary $\mathcal{A}$, the following definition of advantage is negligible in λ.*

$$\mathbf{Adv}^{\mathsf{DLIN}}_{\Gamma,\mathcal{A}} := 2 \cdot \Pr[\mathsf{DLIN} \Rightarrow \mathsf{True}] - 1,$$

where game DLIN *is described in Fig. 2.*

Bilinear Groups of Composite Order

Definition 4. *A composite-order bilinear group generator is an algorithm* $\mathcal{G}_\mathcal{C}$ *that takes as input a security parameter* λ *and outputs a description* $\Gamma = (\mathsf{p}, \mathsf{q}, \mathbb{G}, \mathbb{G}_\mathsf{T}, \mathbf{e}, \mathsf{g})$ *where* $\mathbb{G}$ *and* $\mathbb{G}_\mathsf{T}$ *are groups of order* $\mathsf{n} = \mathsf{pq}$*, where* p *and* q *are independent* λ*-bit primes, with efficiently computable group laws,* g *is a generator of* $\mathbb{G}$ *and* $\mathbf{e}$ *is an efficiently-computable bilinear pairing* $\mathbf{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_\mathsf{T}$.

Subgroups $\mathbb{G}_\mathsf{p} \subset \mathbb{G}$ and $\mathbb{G}_\mathsf{q} \subset \mathbb{G}$ of order p and order q can be generated respectively by $\mathsf{g}_\mathsf{p} = \mathsf{g}^\mathsf{q}$ and $\mathsf{g}_\mathsf{q} = \mathsf{g}^\mathsf{p}$. We recall some important facts regarding these groups:

- $\mathbb{G} = \mathbb{G}_\mathsf{p} \times \mathbb{G}_\mathsf{q}$
- $\mathbf{e}(\mathsf{g}_\mathsf{p}, \mathsf{g}_\mathsf{q}) = \mathbf{e}(\mathsf{g}^\mathsf{q}, \mathsf{g}^\mathsf{p}) = \mathbf{e}(\mathsf{g}, \mathsf{g})^\mathsf{n} = 1$
- $\mathbf{e}(\mathsf{g}_\mathsf{p}, (\mathsf{g}_\mathsf{p})^\mathsf{a} \cdot (\mathsf{g}_\mathsf{q})^\mathsf{b}) = \mathbf{e}(\mathsf{g}_\mathsf{p}, (\mathsf{g}_\mathsf{p})^\mathsf{a}) \cdot \mathbf{e}(\mathsf{g}_\mathsf{p}, (\mathsf{g}_\mathsf{q})^\mathsf{b}) = \mathbf{e}(\mathsf{g}_\mathsf{p}, \mathsf{g}_\mathsf{p})^\mathsf{a}$

Definition 5. *Let* $\Gamma = (\mathsf{p}, \mathsf{q}, \mathbb{G}, \mathbb{G}_\mathsf{T}, \mathbf{e}, \mathsf{g})$ *be the description output by* $\mathcal{G}_\mathcal{C}(\lambda)$ *and* $\Gamma' = (\mathsf{n}, \mathbb{G}, \mathbb{G}_\mathsf{T}, \mathbf{e}, \mathsf{g})$*, where* $\mathsf{n} \leftarrow \mathsf{pq}$*. We say the C3DH assumption holds for description* Γ' *if, for every PPT adversary* $\mathcal{A}$*, the following definition of advantage is negligible in* λ*.*

$$\mathbf{Adv}^{\mathsf{C3DH}}_{\Gamma',\mathcal{A}} := 2 \cdot \Pr[\mathsf{C3DH} \Rightarrow \mathsf{True}] - 1,$$

where game C3DH *is described in Fig. 3.*

Definition 6. *Let* $\Gamma = (\mathsf{p}, \mathsf{q}, \mathbb{G}, \mathbb{G}_\mathsf{T}, \mathbf{e}, \mathsf{g})$ *be the description output by* $\mathcal{G}_\mathcal{C}(\lambda)$ *and* $\Gamma' = (\mathsf{n}, \mathbb{G}, \mathbb{G}_\mathsf{T}, \mathbf{e}, \mathsf{g})$*, where* $\mathsf{n} \leftarrow \mathsf{pq}$*. We say the CDDH assumption holds for description* Γ' *if, for every PPT adversary* $\mathcal{A}$*, the following definition of advantage is negligible in* λ*.*

$$\mathbf{Adv}^{\mathsf{CDDH}}_{\Gamma',\mathcal{A}} := 2 \cdot \Pr[\mathsf{CDDH} \Rightarrow \mathsf{True}] - 1,$$

where game CDDH *is described in Fig. 4.*

procedure Initialize(λ):
$(\mathsf{p}, \mathsf{q}, \mathbb{G}, \mathbb{G}_\mathsf{T}, \mathbf{e}, \mathsf{g}) \leftarrow_\$ \mathcal{G}_\mathcal{C}(\lambda)$
$\mathsf{n} \leftarrow \mathsf{pq}$; $\mathsf{g}_\mathsf{p} \leftarrow \mathsf{g}^\mathsf{q}$; $\mathsf{g}_\mathsf{q} \leftarrow \mathsf{g}^\mathsf{p}$
$\Gamma' \leftarrow (\mathsf{n}, \mathbb{G}, \mathbb{G}_\mathsf{T}, \mathbf{e}, \mathsf{g})$
$\mathsf{X}_1 \leftarrow_\$ \mathbb{G}_\mathsf{q}$; $\mathsf{X}_2 \leftarrow_\$ \mathbb{G}_\mathsf{q}$; $\mathsf{X}_3 \leftarrow_\$ \mathbb{G}_\mathsf{q}$
$\mathsf{a} \leftarrow_\$ \mathbb{Z}_\mathsf{n}$; $\mathsf{b} \leftarrow_\$ \mathbb{Z}_\mathsf{n}$; $\mathsf{c} \leftarrow_\$ \mathbb{Z}_\mathsf{n}$; $\mathsf{R} \leftarrow_\$ \mathbb{G}$
$\mathsf{bit} \leftarrow_\$ \{0, 1\}$
if $\mathsf{bit} = 0$ return
... $(\Gamma', \mathsf{g}_\mathsf{p}, \mathsf{g}_\mathsf{q}, (\mathsf{g}_\mathsf{p})^\mathsf{a}, (\mathsf{g}_\mathsf{p})^\mathsf{b}, \mathsf{X}_1(\mathsf{g}_\mathsf{p})^{\mathsf{ab}}, \mathsf{X}_2(\mathsf{g}_\mathsf{p})^{\mathsf{abc}}, \mathsf{X}_3(\mathsf{g}_\mathsf{p})^\mathsf{c})$
else return
... $(\Gamma', \mathsf{g}_\mathsf{p}, \mathsf{g}_\mathsf{q}, (\mathsf{g}_\mathsf{p})^\mathsf{a}, (\mathsf{g}_\mathsf{p})^\mathsf{b}, \mathsf{X}_1(\mathsf{g}_\mathsf{p})^{\mathsf{ab}}, \mathsf{X}_2(\mathsf{g}_\mathsf{p})^{\mathsf{abc}}, \mathsf{R})$

procedure Finalize(bit'):
if $\mathsf{bit} = \mathsf{bit}'$ return True
else return False

Fig. 3. Game C3DH

procedure Initialize(λ):
$(\mathsf{p}, \mathsf{q}, \mathbb{G}, \mathbb{G}_\mathsf{T}, \mathbf{e}, \mathsf{g}) \leftarrow_\$ \mathcal{G}_\mathcal{C}(\lambda)$
$\mathsf{n} \leftarrow \mathsf{pq}$; $\mathsf{g}_\mathsf{p} \leftarrow \mathsf{g}^\mathsf{q}$; $\mathsf{g}_\mathsf{q} \leftarrow \mathsf{g}^\mathsf{p}$
$\Gamma' \leftarrow (\mathsf{n}, \mathbb{G}, \mathbb{G}_\mathsf{T}, \mathbf{e}, \mathsf{g})$
$\mathsf{X}_1 \leftarrow_\$ \mathbb{G}_\mathsf{q}$; $\mathsf{X}_2 \leftarrow_\$ \mathbb{G}_\mathsf{q}$; $\mathsf{X}_3 \leftarrow_\$ \mathbb{G}_\mathsf{q}$
$\mathsf{a} \leftarrow_\$ \mathbb{Z}_\mathsf{n}$; $\mathsf{b} \leftarrow_\$ \mathbb{Z}_\mathsf{n}$; $\mathsf{R} \leftarrow_\$ \mathbb{G}$
$\mathsf{bit} \leftarrow_\$ \{0, 1\}$
if $\mathsf{bit} = 0$ return
... $(\Gamma', \mathsf{g}_\mathsf{p}, \mathsf{g}_\mathsf{q}, \mathsf{X}_1(\mathsf{g}_\mathsf{p})^\mathsf{a}, \mathsf{X}_2(\mathsf{g}_\mathsf{p})^\mathsf{b}, \mathsf{X}_3(\mathsf{g}_\mathsf{p})^{\mathsf{ab}})$
else return
... $(\Gamma', \mathsf{g}_\mathsf{p}, \mathsf{g}_\mathsf{q}, \mathsf{X}_1(\mathsf{g}_\mathsf{p})^\mathsf{a}, \mathsf{X}_2(\mathsf{g}_\mathsf{p})^\mathsf{b}, \mathsf{R})$

procedure Finalize(bit'):
if $\mathsf{bit} = \mathsf{bit}'$ return True
else return False

Fig. 4. Game CDDH

In game C3DH, adversary is given a tuple $(\Gamma', \mathsf{g_p}, \mathsf{g_q}, (\mathsf{g_p})^{\mathsf{a}}, (\mathsf{g_p})^{\mathsf{b}}, \mathsf{X}_1(\mathsf{g_p})^{\mathsf{ab}}, \mathsf{X}_2(\mathsf{g_p})^{\mathsf{abc}}, \mathsf{Z})$ and has to decide whether $\mathsf{Z} = \mathsf{X}_3(\mathsf{g_p})^{\mathsf{c}}$, for some $\mathsf{X}_3 \in \mathbb{G}_{\mathsf{q}}$. For convenience, we rewrite this as $(\Gamma', \mathsf{g_p}, \mathsf{g_q}, (\mathsf{g_p})^{\mathsf{a}}, (\mathsf{g_p})^{\mathsf{b}}, \mathsf{X}_1(\mathsf{g_p})^{\mathsf{ab}}, \mathsf{Y}, \mathsf{X}_3(\mathsf{g_p})^{\mathsf{c}})$, where Y is either $\mathsf{X}_2(\mathsf{g_p})^{\mathsf{abc}}$ or random in $\mathbb{G}$. Now, notice that $(\Gamma', \mathsf{g_p}, \mathsf{g_q}, \mathsf{X}_1(\mathsf{g_p})^{\mathsf{ab}}, \mathsf{X}_3(\mathsf{g_p})^{\mathsf{c}}, \mathsf{Y})$ is a CDDH tuple. Therefore, CDDH is a weaker assumption than C3DH.

2.2 Anonymous Identity-Based Encryption

An IBE scheme $\Pi = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ is specified by four polynomial-time algorithms associated with a message space $\mathcal{M}$ and an identity space $\mathcal{I}$.

- $\mathsf{Setup}(\lambda)$: On input the security parameter λ, this algorithm returns a master secret key msk and public parameters pp.
- $\mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id})$: On input public parameters pp, a master secret key msk and an identity $\mathsf{id} \in \mathcal{I}$, this algorithm outputs a secret key sk.
- $\mathsf{Enc}(\mathsf{pp}, \mathsf{m}, \mathsf{id})$: On input public parameters pp, a message $\mathsf{m} \in \mathcal{M}$ and an identity $\mathsf{id} \in \mathcal{I}$, this algorithm outputs a ciphertext c.
- $\mathsf{Dec}(\mathsf{pp}, \mathsf{c}, \mathsf{sk})$: On input public parameters pp, a ciphertext c and a secret key sk, this algorithm outputs either a message m or a failure symbol $\perp$.

The correctness of an IBE scheme requires that decryption reverses encryption, i.e., for any $\lambda \in \mathbb{N}$, any $(\mathsf{msk}, \mathsf{pp}) \leftarrow_{\$} \mathsf{Setup}(\lambda)$, any $\mathsf{id} \in \mathcal{I}$, any $\mathsf{m} \in \mathcal{M}$, we have that $\mathsf{Dec}(\mathsf{pp}, \mathsf{Enc}(\mathsf{pp}, \mathsf{m}, \mathsf{id}), \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id})) = \mathsf{m}$.

The standard notions of security for IBE are *anonymity* and *semantic security*. Intuitively, *anonymity* requires that ciphertexts conceal the identity and *semantic security* requires that ciphertexts conceal the message. We omit the formal definitions in this version due to space limitations. These properties lead to *semantic security* and *computational consistency*, respectively, in PEKS, after applying the black-box transformation described in [1].

3 Security Definitions

In this section, we formulate the notion of *Weak Key Unlinkability* for IBE, which leads to Nishioka's *Search Pattern Privacy* model for PEKS [14], after the black-box transformation from IBE to PEKS [1]. We then strengthen the model by allowing the adversary to be challenged with multiple secret keys, instead of just two. We refer to this new model as *Strong Key Unlinkability*. The resulting "dual" property for PEKS allows the adversary to be challenged with multiple trapdoors, which better reflects real-world scenarios. We then compare the new notions of security introduced here with those introduced by Boneh et al. in [9], and show that the two are orthogonal. Finally, we show that an easy and natural transformation from Strong Key Unlinkability to a more generalized definition, where the adversary is allowed to choose a joint probability distribution from which identities are sampled - instead of being sampled uniformly at random from the identity space - exists, as long as the adversary's choice does not depend on the public parameters of the scheme.

3.1 Key Unlinkability for IBE

Key Unlinkability models for IBE require that the size of the identity space is at least $\omega(\log \lambda)$, where λ is the security parameter of the scheme.

Definition 7. *An IBE scheme Π, associated with a non-polynomial size identity space $\mathcal{I}$, has Weak Key Unlinkability if, for every legitimate PPT adversary $\mathcal{A}$, the following definition of advantage is negligible in λ*

$$\mathbf{Adv}^{\mathsf{WEAK\text{-}KEY\text{-}UNLINK}}_{\Pi,\mathcal{A}}(\lambda) := 2 \cdot \Pr[\mathsf{WEAK\text{-}KEY\text{-}UNLINK}(\lambda) \Rightarrow \mathsf{True}] - 1,$$

where game WEAK-KEY-UNLINK *is described in Fig. 5.*

Definition 8. *An IBE scheme Π, associated with a non-polynomial size identity space $\mathcal{I}$, has Strong Key Unlinkability if, for every legitimate PPT adversary $\mathcal{A}$, the following definition of advantage is negligible in λ*

$$\mathbf{Adv}^{\mathsf{STRONG\text{-}KEY\text{-}UNLINK}}_{\Pi,\mathcal{A}}(\lambda) := 2 \cdot \Pr[\mathsf{STRONG\text{-}KEY\text{-}UNLINK}(\lambda) \Rightarrow \mathsf{True}] - 1,$$

where game STRONG-KEY-UNLINK *is described in Fig. 6.*

procedure Initialize(λ):
$(\mathsf{msk}, \mathsf{pp}) \leftarrow_\$ \mathsf{Setup}(\lambda)$
$\mathsf{bit} \leftarrow_\$ \{0,1\}$
$\mathsf{id}_0 \leftarrow_\$ \mathcal{I}$
$\mathsf{id}_1 \leftarrow_\$ \mathcal{I}$
$\mathsf{sk}_0 \leftarrow_\$ \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}_0)$
$\mathsf{sk}_1 \leftarrow_\$ \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}_{\mathsf{bit}})$
return $(\mathsf{pp}, \mathsf{sk}_0, \mathsf{sk}_1)$

procedure Extract(id):
$\mathsf{sk}_{\mathsf{id}} \leftarrow_\$ \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id})$
return $\mathsf{sk}_{\mathsf{id}}$

procedure Finalize(bit'):
return $(\mathsf{bit} = \mathsf{bit}')$

Fig. 5. Game WEAK-KEY-UNLINK

procedure Initialize(λ):
$(\mathsf{msk}, \mathsf{pp}) \leftarrow_\$ \mathsf{Setup}(\lambda)$
$\mathsf{bit} \leftarrow_\$ \{0,1\}$
$\mathsf{list}_{\mathsf{id}} \leftarrow []$
$\mathsf{list}_{\mathsf{sk}} \leftarrow []$
return pp

procedure Extract(id):
$\mathsf{sk} \leftarrow_\$ \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id})$
return tp

procedure Finalize(bit'):
return $(\mathsf{bit} = \mathsf{bit}')$

procedure Challenge$(\mathsf{list}_0, \mathsf{list}_1)$:
$\mathsf{L} \leftarrow |\mathsf{list}_0|$
for i in {1..L}
... get id for $\mathsf{list}_{\mathsf{bit}}[\mathsf{i}]$ from $\mathsf{list}_{\mathsf{id}}$
... if $\mathsf{id} = \perp$
... ... $\mathsf{id} \leftarrow_\$ \mathcal{I}$
... ... $\mathsf{list}_{\mathsf{id}} \leftarrow (\mathsf{list}_{\mathsf{bit}}[\mathsf{i}], \mathsf{id}) : \mathsf{list}_{\mathsf{id}}$
... $\mathsf{list}_{\mathsf{sk}}[\mathsf{i}] \leftarrow_\$ \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id})$
return $\mathsf{list}_{\mathsf{sk}}$

Fig. 6. Game STRONG-KEY-UNLINK. Adversary is legitimate if it only calls Challenge once with $|\mathsf{list}_0| = |\mathsf{list}_1|$.

3.2 Function Privacy for IBE: An Independent Security Notion

Recently, Boneh, Raghunathan and Segev [9] put forward two security notions, of different strength, for IBE, inspired by the security definition given for deterministic encryption in [3]: *Function Privacy* and *Enhanced Function Privacy*. These notions ask that "decryption keys reveal essentially no information on their corresponding identities, beyond the absolute minimum necessary". In both definitions, the adversary is first given the public parameters and then interacts with a Real-or-Random function privacy oracle, which takes as input an adversarially-chosen joint probability distribution – represented as a circuit – for random variables $\mathsf{X}_1, \mathsf{X}_2, ..., \mathsf{X}_\mathsf{L}$ defined over the identity space $\mathcal{I}$, and outputs L secret keys either for identities sampled from the given joint probability distribution or for independent and uniformly distributed identities over $\mathcal{I}$.

An adversary is legitimate if, for every $\mathsf{i} \in \{1..\mathsf{L}\}$ and every $\mathsf{x}_1, ..., \mathsf{x}_\mathsf{i} \in \mathcal{I}$, it holds that: $\mathbf{H}_\infty(\mathsf{X}_\mathsf{i}|_{\mathsf{X}_1=\mathsf{x}_1,\ ...,\mathsf{X}_{\mathsf{i}-1}=\mathsf{x}_{\mathsf{i}-1}}) = -\log(\max \Pr[\mathsf{X}_\mathsf{i} = \mathsf{x}_\mathsf{i}|_{\mathsf{X}_1=\mathsf{x}_1,...,\mathsf{X}_{\mathsf{i}-1}=\mathsf{x}_{\mathsf{i}-1}}]) \geq \omega(\log \lambda)$. Put differently, the chosen joint probability distribution for $(\mathsf{X}_1, ..., \mathsf{X}_\mathsf{L})$ has to be such that every random variable X_i is *sufficiently unpredictable*, even if every random variable $\mathsf{X}_{\mathsf{j}<\mathsf{i}}$ has been fixed. To discard exhaustive searches, a *conditional min-entropy* $\mathbf{H}_\infty(\mathsf{X}_\mathsf{i}|_{\mathsf{X}_1=\mathsf{x}_1,...,\mathsf{X}_{\mathsf{i}-1}})$ of at least $\omega(\log \lambda)$ bits is required[2]. The *Enhanced* model provides the adversary with an extra function-privacy encryption oracle capable of encrypting adversarially-chosen messages under the identities sampled by Real-or-Random oracle. Formal definitions can be found in [9].

We first remark that Key Unlinkability and Function Privacy security models are essentially different in the way the challenger samples ids: in the former ids are sampled uniformly from the id space, whereas in the latter model ids may be sampled from an adversarial-chosen joint probability distribution, with (possibly) non-uniform random variables, but also high min-entropy requirements. In the following subsections we provide counterexamples to show that Function Privacy (both *Non-enhanced* and *Enhanced*) and Key Unlinkability (both *Weak* and *Strong*) are independent security notions. Meaningful counterexamples follow. For a quick overview, Figure 7 states the relations between Weak Key Unlinkability, Strong Key Unlinkability, Function Privacy and Enhanced Function Privacy security notions.

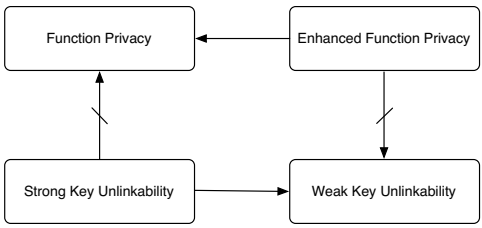

Fig. 7. Relations between Key Unlinkability and Function Privacy security notions

We stress that even Enhanced Function Privacy fails to capture the security guarantees of Weak Key Unlinkability. In practice, transforming an anonymous IBE with Enhanced Function Privacy to PEKS (according to the transformation described in Section 2) results in no guarantee that the service provider will not be able to find search patterns in the users' trapdoors.

Enhanced Function Privacy $\not\Rightarrow$ Weak Key Unlinkability. Consider $F : \{0,1\}^\lambda \times \mathcal{I} \rightarrow \{0,1\}^\lambda$ to be a secure PRF. We denote by $f \leftarrow_\$ F$ the operation: $\mathsf{k} \leftarrow_\$ \{0,1\}^\lambda; f \leftarrow F(\mathsf{k}, \cdot)$. Let $\Pi = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ be an enhanced function-private IBE. From Π we can construct Π', where Π' is still enhanced function-private but definitely not weak key-unlinkable. We do so by simply modifying the extraction algorithm and appending to each secret key

[2] The *minimal* unpredictability requirement of $\omega(\log \lambda)$ bits has only been achieve later in [10]. Schemes in [9] have only been proven secure for highly unpredictable identities with min-entropy of $\lambda + \omega(\log \lambda)$.

the result of a PRF on id. More precisely, $\Pi' = (\mathsf{Setup}', \mathsf{Extract}', \mathsf{Enc}', \mathsf{Dec}')$ is constructed as follows:

- $\mathsf{Setup}'(\lambda) : (\mathsf{msk}, \mathsf{pp}) \leftarrow_\$ \mathsf{Setup}(\lambda)$; $f \leftarrow_\$ F$; return $((\mathsf{msk}, f), \mathsf{pp})$.
- $\mathsf{Extract}'(\mathsf{msk}, \mathsf{id}) : \mathsf{sk} \leftarrow_\$ \mathsf{Extract}(\mathsf{msk}, \mathsf{id}); \mathsf{sk}' \leftarrow (\mathsf{sk}, f(\mathsf{id}))$; return sk'.
- $\mathsf{Enc}'(\mathsf{pp}, \mathsf{m}, \mathsf{id}) : \mathsf{c} \leftarrow_\$ \mathsf{Enc}(\mathsf{pp}, \mathsf{m}, \mathsf{id})$; return c.
- $\mathsf{Dec}'(\mathsf{pp}, \mathsf{c}, \mathsf{id}, \mathsf{sk}') : (\mathsf{sk}, \mathsf{y}) \leftarrow \mathsf{sk}'$; $\mathsf{m} \leftarrow \mathsf{Dec}(\mathsf{pp}, \mathsf{c}, \mathsf{id}, \mathsf{sk})$; return m.

Informally, since f is unknown to the adversary, the adversary cannot choose distributions depending on f. Furthermore, F is a secure PRF, so no information on id can be acquired. Therefore, Π' is still an enhanced function-private IBE. But, because f is deterministic, it is trivial to identify with overwhelming probability if two keys have been extracted from the same identity.

Strong Key Unlinkability $\not\Longrightarrow$ Function Privacy. Again, we show this by counterexample. Let $\Pi = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ be a strong key-unlinkable IBE associated with id space $\mathcal{I} = \{0,1\}^{2\lambda}$. We build $\Pi' = (\mathsf{Setup}', \mathsf{Extract}', \mathsf{Enc}', \mathsf{Dec}')$ based on Π as follows:

- $\mathsf{Setup}'(\lambda) : (\mathsf{msk}, \mathsf{pp}) \leftarrow_\$ \mathsf{Setup}(\lambda)$; return $(\mathsf{msk}, \mathsf{pp})$.
- $\mathsf{Extract}'(\mathsf{msk}, \mathsf{id}) : \mathsf{sk} \leftarrow_\$ \mathsf{Extract}(\mathsf{msk}, \mathsf{id})$; if $\mathsf{id} \in \{0,1\}^\lambda 0^\lambda$ then $\mathsf{sk}' \leftarrow (\mathsf{sk}||0)$ else $\mathsf{sk}' \leftarrow (\mathsf{sk}||1)$; return sk'.
- $\mathsf{Enc}'(\mathsf{pp}, \mathsf{m}, \mathsf{id}) : \mathsf{c} \leftarrow_\$ \mathsf{Enc}(\mathsf{pp}, \mathsf{m}, \mathsf{id})$ return c.
- $\mathsf{Dec}'(\mathsf{pp}, \mathsf{c}, \mathsf{id}, \mathsf{sk}') : (\mathsf{sk}||\mathsf{b}) \leftarrow \mathsf{sk}'; \mathsf{m} \leftarrow \mathsf{Dec}(\mathsf{pp}, \mathsf{c}, \mathsf{id}, \mathsf{sk})$; return m.

In our counterexample scheme Π', we put a mark in keys for identities whose last λ bits are 0, by appending a 0 to the key (otherwise, 1 is appended). Since the subset containing these identities – let us call it $\mathcal{U}$ – is much smaller than the identity space $\mathcal{I}$, identities uniformly sampled from $\mathcal{I}$ are very unlikely to be in $\mathcal{U}$, and thus to possess the mark. In fact, this only happens with probability $\Pr = \frac{2^\lambda}{2^{2\lambda}} = \frac{1}{2^\lambda}$, which is a negligible function in the security parameter λ. Strong Key Unlinkability is therefore preserved in Π'. However, $\mathcal{U}$ is big enough so that the unpredictability of an id uniformly sampled from $\mathcal{U}$ is high. By choosing to be challenged on a random variable X that selects any element in $\mathcal{U}$ with probability $\frac{1}{2^\lambda}$ and any element in $\{\mathsf{x} \in \mathcal{I} : \mathsf{x} \notin \mathcal{U}\}$ with zero probability, an adversary could trivially win the function-privacy game, with overwhelming probability, just by looking into the key's mark. Also notice that the condition $\mathbf{H}_\infty(\mathsf{X}) > \omega(\log \lambda)$ is satisfied. Generically, we can conclude that a strong key-unlinkable scheme is not necessarily function-private secure.

3.3 Adversarially-Chosen Joint Probability Distributions of Keywords

In security game Strong Key Unlinkability [Fig. 6], identities are sampled uniformly at random from the identity space, as opposed to from a (possibly non-uniform) adversarially-chosen joint probability distribution. The latter approach was used by Boneh et al. [9] to form the challenge in Function Privacy security models. In most real-world applications of PEKS, keywords are not chosen uniformly from the

keyword space. Therefore, it is important to discuss the choice of our model and the impact of generalizing it to deal with adversarially-chosen distributions.

The full version of [9] proposes a generic method for transforming any IBE scheme into an IBE scheme which achieves a *weaker* form of Enhanced Function Privacy, where the adversary is not allowed to choose a joint probability distribution (from which identities are sampled for the challenge) that depends on the public parameters of the scheme. In fact, the challenger only provides the public parameters *after* the joint probability distribution is fixed by the adversary. This relaxation results in a definition denoted *Non-Adaptive* Enhanced Function Privacy.

Adopting the same strategy as [9,3], we strengthen our model by allowing the adversary to choose a joint probability distribution from which identities are sampled, instead of lists defining equality relations between identities. The environment of the game becomes exactly that of Non-Enhanced Function Privacy defined in [9] (and described here, in Subsection 3.2), but the unpredictability requirements on what constitutes a legitimate joint probability distribution $\mathbb{X} = \{\mathsf{X}_1, ..., \mathsf{X}_\mathsf{L}\}$ are relaxed to $\mathbf{H}_\infty(\mathsf{X}_\mathsf{i}) \geq \omega(\log \lambda)$, for every $\mathsf{i} \in \{1..\mathsf{L}\}$. Public parameters can be provided before or after the adversary fixes $\mathbb{X}$, resulting in two models of different strengths. We refer to the model where the adversary fixes a joint probability distribution (with possibly non-uniform random variables) from which the challenger samples the identities *after* (resp. *before*) receiving the public parameters as *Adaptive* (resp. *Non-Adaptive*) *Key Unlinkability*.

Definition 9. *An IBE scheme Π, associated with a non-polynomial size identity space $\mathcal{I}$, has Non-Adaptive Key Unlinkability if, for every legitimate PPT adversary $\mathcal{A}$, the following definition of advantage is negligible in λ*

$$\mathbf{Adv}^{\mathsf{KEY\text{-}UNLINK}}_{\Pi,\mathcal{A}_{\mathsf{nonadaptive}}}(\lambda) := 2{\cdot}\Pr[\mathsf{KEY\text{-}UNLINK}(\lambda, \mathsf{mode} = \text{“non-adaptive”}) \Rightarrow \mathsf{True}]{-}1,$$

where game KEY-UNLINK *is described in Fig. 8.*

procedure Initialize(λ, mode):	**procedure Challenge**$(\mathbb{X} = \{\mathsf{X}_1, ..., \mathsf{X}_\mathsf{L}\})$:
$(\mathsf{msk}, \mathsf{pp}) \leftarrow_\$ \mathsf{Setup}(\lambda)$	if $\mathsf{bit} = 0$
$\mathsf{bit} \leftarrow_\$ \{0, 1\}$	... $(\mathsf{id}_1, ..., \mathsf{id}_\mathsf{L}) \leftarrow_\$ \mathbb{X}$
$\mathsf{list} \leftarrow []$	if $\mathsf{bit} = 1$
if $\mathsf{mode} =$ “adaptive” return pp	... $(\mathsf{id}_1, ..., \mathsf{id}_\mathsf{L}) \leftarrow_\$ \mathcal{I}^\mathsf{L}$
	for $\mathsf{i} \in \{1..\mathsf{L}\}$
	... $\mathsf{list}[\mathsf{i}] \leftarrow_\$ \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}_\mathsf{i})$
	return $(\mathsf{list}, \mathsf{pp})$
procedure Extract(id):	**procedure Finalize**(bit'):
$\mathsf{sk} \leftarrow_\$ \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id})$	return $(\mathsf{bit} = \mathsf{bit}')$
return tp	

Fig. 8. Game KEY-UNLINK. $\mathbb{X} = \{\mathsf{X}_1, ..., \mathsf{X}_\mathsf{L}\}$ is a joint probability distribution with L random variables defined over the identity space $\mathcal{I}$. Adversary is legitimate if $\mathbf{H}_\infty(\mathsf{X}_\mathsf{i}) \geq \omega(\log \lambda)$, for every $\mathsf{i} \in \{1..\mathsf{L}\}$.

Definition 10. *An IBE scheme Π, associated with a non-polynomial size identity space $\mathcal{I}$, has Adaptive Key Unlinkability if, for every legitimate PPT adversary $\mathcal{A}$, the following definition of advantage is negligible in λ*

$$\mathbf{Adv}^{\mathsf{KEY\text{-}UNLINK}}_{\Pi,\mathcal{A}_{\mathsf{adaptive}}}(\lambda) := 2 \cdot \Pr[\mathsf{KEY\text{-}UNLINK}(\lambda, \mathsf{mode} = \text{“}\mathsf{adaptive}\text{”}) \Rightarrow \mathsf{True}] - 1,$$

where game KEY-UNLINK *is described in Fig. 8.*

Remark. The joint probability distribution $\mathbb{X} = \{\mathsf{X}_1, \mathsf{X}_2\}$ such that $\Pr[\mathsf{X}_2 = \mathsf{x}_1] = 1$ is legitimate for Adaptive (and Non-Adaptive) Key Unlinkability, as long as $\mathbf{H}_\infty(\mathsf{X}_1) \geq \omega(\log \lambda)$. In particular, if X_1 is a uniform random variable in $\mathcal{I}$, then the game becomes that of Weak Key Unlinkability [Fig. 5]. However, as expected, $\mathbb{X}$ is not a legitimate joint probability distribution for Function Privacy (Enhanced or Non-Enhanced, Adaptive or Non-Adaptive).

Towards Non-Adaptive Key Unlinkability. We now show that there is an easy and natural transformation from Strong Key Unlinkability to Non-Adaptive Key Unlinkability. Let $\Pi = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ be an IBE scheme, associated with message space $\mathcal{M}$ and identity space $\mathcal{I}$, and let $\mathcal{H} : \mathcal{I}' \to \mathcal{I}$ be a family of hash functions. We construct an IBE scheme $\Pi' = (\mathsf{Setup}', \mathsf{Extract}', \mathsf{Enc}', \mathsf{Dec}')$, associated with message space $\mathcal{M}$ and identity space $\mathcal{I}'$, as follows:

- $\mathsf{Setup}'(\lambda) : (\mathsf{msk}, \mathsf{pp}) \leftarrow_\$ \mathsf{Setup}(\lambda)$; $\mathsf{H} \leftarrow_\$ \mathcal{H}$; return $(\mathsf{msk}, (\mathsf{pp}, \mathsf{H}))$.
- $\mathsf{Extract}'(\mathsf{msk}, \mathsf{id}') : \mathsf{id} \leftarrow \mathsf{H}(\mathsf{id}')$; $\mathsf{sk} \leftarrow_\$ \mathsf{Extract}(\mathsf{msk}, \mathsf{id})$; return sk.
- $\mathsf{Enc}'((\mathsf{pp}, \mathsf{H}), \mathsf{m}, \mathsf{id}') : \mathsf{id} \leftarrow \mathsf{H}(\mathsf{id}')$; $\mathsf{c} \leftarrow_\$ \mathsf{Enc}(\mathsf{pp}, \mathsf{m}, \mathsf{id})$; return c.
- $\mathsf{Dec}'((\mathsf{pp}, \mathsf{H}), \mathsf{c}, \mathsf{id}, \mathsf{sk}') : \mathsf{id} \leftarrow \mathsf{H}(\mathsf{id}')$; $\mathsf{m} \leftarrow \mathsf{Dec}(\mathsf{pp}, \mathsf{c}, \mathsf{id}, \mathsf{sk})$; return m.

Lemma 1. *If $|\mathcal{I}'| \geq |\mathcal{I}| \geq 2^{\omega(\log \lambda)}$ and IBE scheme Π has Strong Key Unlinkability, then IBE scheme Π' has Non-Adaptive Key Unlinkability, in the random oracle model.*

Proof. Let $\mathcal{A}$ be a legitimate adversary against Non-Adaptive Key Unlinkability, and let $\mathbb{X} = \{\mathsf{X}_1, ..., \mathsf{X}_\mathsf{L}\}$ be the joint probability distribution that $\mathcal{A}$ chooses for the challenge. We recall that a legitimate adversary is required to choose $\mathbb{X}$ such that $\forall\ \mathsf{X}_\mathsf{i} \in \mathbb{X}, \mathbf{H}_\infty(\mathsf{X}_\mathsf{i}) \geq \omega(\log \lambda)$, where λ is the security parameter of Π. Game_0 is the original Non-Adaptive Key Unlinkability game described above, instantiated with IBE scheme Π'. In Game_1, H is modeled as a random oracle. $(\mathsf{id}'_1, ..., \mathsf{id}'_\mathsf{L}) \leftarrow_\$ \{\mathsf{X}_1, ..., \mathsf{X}_\mathsf{L}\}$ forms a list of bit-strings. A simulator $\mathcal{S}$ could construct the challenge of Game_1 by setting $\mathsf{list}_0 = (\mathsf{id}'_1, ..., \mathsf{id}'_\mathsf{L}) \leftarrow_\$ \{\mathsf{X}_1, ..., \mathsf{X}_\mathsf{L}\}$ and list_1 with L different bit-strings, and querying the challenge procedure of $\mathsf{STRONG\text{-}KEY\text{-}UNLINK}_{\Pi,\mathcal{S}}$ with $(\mathsf{list}_0, \mathsf{list}_1)$. The result is a well-formed tuple of L secret keys, and $\mathcal{A}$'s final guess can be forward to $\mathsf{STRONG\text{-}KEY\text{-}UNLINK}_{\Pi,\mathcal{S}}$. Simulator $\mathcal{S}$ perfectly mimics the environment of Game_1, unless $\mathcal{A}$ queries the hash value of any id'_i, in which case the simulation aborts. However, this event only happens negligible probability. Therefore, we have that $\mathbf{Adv}^{\mathsf{KEY\text{-}UNLINK}}_{\Pi',\mathcal{A}_{\mathsf{nonadaptive}}}(\lambda) \leq \mathbf{Adv}^{\mathsf{STRONG\text{-}KEY\text{-}UNLINK}}_{\Pi,\mathcal{S}}(\lambda) + \frac{\mathsf{q} \cdot \mathsf{L}}{2^{\omega(\log \lambda)}}$, where q is the number of queries $\mathcal{A}$ asks to the random oracle. □

Most IBE schemes, including the one introduced in this paper later on, only make use of the hash value of identities (instead of the identities themselves). Thus, the simplicity of Strong Key Unlinkability does not come at the expense of the model's security meaning. From a theoretical point of view, it seems interesting (but we leave it as future work) to investigate the construction of IBE schemes that achieve Key Unlinkability against *adaptive* adversaries. In practice, for what concerns PEKS, it seems reasonable to assume that keywords will not depend on the public parameters of the scheme, and, in particular, on the values output by the hash function.

4 From Weak to Strong Key Unlinkability

A scheme with Weak Key Unlinkability. We construct a new anonymous IBE scheme with Weak Key Unlinkability, based on the anonymous IBE scheme of Boyen and Waters [12]. Our scheme relies on a bilinear group description Γ of prime order. To eliminate the selective-ID constraint, we replace identities with their hash values and model the hash function as a random oracle. Furthermore, we simplify the resulted scheme by removing two group elements from the public parameters and from private keys, and obtain the final scheme in Fig. 9. Compared with the original scheme, our scheme also saves two exponentiations in the key-extraction and encryption algorithms, and saves two pairing computations in the decryption algorithm. Our scheme preserves *anonymity* and *semantic security* properties, provided that the hash function H, selected from a family of hash functions $\mathcal{H} : \mathcal{I} \to \mathbb{G}$, is modeled as a random oracle. Added to this, the scheme also has the Weak Key Unlinkability property.

$\underline{\mathsf{Setup}(\lambda)}$:	$\underline{\mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id})}$:	$\underline{\mathsf{Enc}(\mathsf{pp}, \mathsf{m}, \mathsf{id})}$:	$\underline{\mathsf{Dec}(\mathsf{pp}, \mathsf{c}, \mathsf{id}, \mathsf{sk}_{\mathsf{id}})}$:
$\Gamma \leftarrow_\$ \mathcal{G}_{\mathcal{P}}(\lambda)$	$\mathsf{r} \leftarrow_\$ \mathbb{Z}_\mathsf{p}$	$\mathsf{s}, \mathsf{s}_1 \leftarrow_\$ \mathbb{Z}_\mathsf{p}^2$	$(\Gamma, \Omega, \mathsf{v}_1, \mathsf{v}_2, \mathsf{H}) \leftarrow \mathsf{pp}$
$(\mathsf{p}, \mathbb{G}, \mathbb{G}_\mathsf{T}, \mathbf{e}, \mathsf{g}) \leftarrow \Gamma$	$(\mathsf{w}, \mathsf{t}_1, \mathsf{t}_2) \leftarrow \mathsf{msk}$	$(\Gamma, \Omega, \mathsf{v}_1, \mathsf{v}_2, \mathsf{H}) \leftarrow \mathsf{pp}$	$(\mathsf{p}, \mathbb{G}, \mathbb{G}_\mathsf{T}, \mathbf{e}, \mathsf{g}) \leftarrow \Gamma$
$\mathsf{w}, \mathsf{t}_1, \mathsf{t}_2 \leftarrow_\$ \mathbb{Z}_\mathsf{p}^3$	$(\Gamma, \Omega, \mathsf{v}_1, \mathsf{v}_2, \mathsf{H}) \leftarrow \mathsf{pp}$	$(\mathsf{p}, \mathbb{G}, \mathbb{G}_\mathsf{T}, \mathbf{e}, \mathsf{g}) \leftarrow \Gamma$	$(\mathsf{d}_0, \mathsf{d}_1, \mathsf{d}_2) \leftarrow \mathsf{sk}_{\mathsf{id}}$
$\Omega \leftarrow \mathbf{e}(\mathsf{g}, \mathsf{g})^{\mathsf{t}_1 \mathsf{t}_2 \mathsf{w}}$	$(\mathsf{p}, \mathbb{G}, \mathbb{G}_\mathsf{T}, \mathbf{e}, \mathsf{g}) \leftarrow \Gamma$	$\mathsf{h} \leftarrow \mathsf{H}(\mathsf{id})$	$(\hat{\mathsf{c}}, \mathsf{c}_0, \mathsf{c}_1, \mathsf{c}_2) \leftarrow \mathsf{c}$
$\mathsf{v}_1 \leftarrow \mathsf{g}^{\mathsf{t}_1}$	$\mathsf{h} \leftarrow \mathsf{H}(\mathsf{id})$	$\hat{\mathsf{c}} \leftarrow \Omega^\mathsf{s} \mathsf{m}$	$\mathsf{e}_0 \leftarrow \mathbf{e}(\mathsf{c}_0, \mathsf{d}_0)$
$\mathsf{v}_2 \leftarrow \mathsf{g}^{\mathsf{t}_2}$	$\mathsf{d}_0 \leftarrow \mathsf{g}^{\mathsf{r}\mathsf{t}_1\mathsf{t}_2}$	$\mathsf{c}_0 \leftarrow \mathsf{h}^\mathsf{s}$	$\mathsf{e}_1 \leftarrow \mathbf{e}(\mathsf{c}_1, \mathsf{d}_1)$
$\mathsf{H} \leftarrow_\$ \mathcal{H} : \mathcal{I} \to \mathbb{G}$	$\mathsf{d}_1 \leftarrow \mathsf{g}^{-\mathsf{w}\mathsf{t}_2} \cdot \mathsf{h}^{-\mathsf{r}\mathsf{t}_2}$	$\mathsf{c}_1 \leftarrow \mathsf{v}_1^{\mathsf{s}-\mathsf{s}_1}$	$\mathsf{e}_2 \leftarrow \mathbf{e}(\mathsf{c}_2, \mathsf{d}_2)$
$\mathsf{pp} \leftarrow (\Gamma, \Omega, \mathsf{v}_1, \mathsf{v}_2, \mathsf{H})$	$\mathsf{d}_2 \leftarrow \mathsf{g}^{-\mathsf{w}\mathsf{t}_1} \cdot \mathsf{h}^{-\mathsf{r}\mathsf{t}_1}$	$\mathsf{c}_2 \leftarrow \mathsf{v}_2^{\mathsf{s}_1}$	$\mathsf{m} \leftarrow \hat{\mathsf{c}} \cdot \mathsf{e}_0 \cdot \mathsf{e}_1 \cdot \mathsf{e}_2$
$\mathsf{msk} \leftarrow (\mathsf{w}, \mathsf{t}_1, \mathsf{t}_2)$	$\mathsf{sk}_{\mathsf{id}} \leftarrow (\mathsf{d}_0, \mathsf{d}_1, \mathsf{d}_2)$	$\mathsf{c} \leftarrow (\hat{\mathsf{c}}, \mathsf{c}_0, \mathsf{c}_1, \mathsf{c}_2)$	return m
return $(\mathsf{msk}, \mathsf{pp})$	return $\mathsf{sk}_{\mathsf{id}}$	return c	

Fig. 9. Anonymous IBE scheme Π with Weak Key Unlinkability

Theorem 1. *IBE scheme Π [Fig. 9] is semantically secure, in the random oracle model, assuming DBDH is intractable [Definition 2].*

Theorem 2. *IBE scheme Π [Fig. 9] is anonymous, in the random oracle model, assuming DBDH and DLIN are intractable [Definitions 2 and 3].*

We omit the proofs of Theorem 1 and Theorem 2 in this version due to space limitations.

Theorem 3 (Appendix A). *IBE scheme Π [Fig. 9] has the Weak Key Unlinkability property [Definition 7], in the random oracle model, assuming DLIN is intractable [Definition 3].*

Weak Key Unlinkability $\not\Longrightarrow$ Strong Key Unlinkability. Standard real-or-random definitions for public-key encryption model the encryption of a single plaintext. These definitions are equivalent (with some loss in tightness) to those allowing an adversary to acquire multiple encryptions, which can be shown by applying the hybrid argument from [2]. One might be tempted to think that the same hybrid argument also applies to Weak Key Unlinkability model. However, this argument does *not* apply, since we can show that an adversary can still easily distinguish patterns when more than two keys are issued with scheme Π [Fig. 9].

Suppose that an adversary is asked to distinguished between tuples of the form $(\mathsf{Extract}(\mathsf{id}_0), \mathsf{Extract}(\mathsf{id}_0), \mathsf{Extract}(\mathsf{id}_0))$, where the three secret keys are extracted from the same id, from those of the form $(\mathsf{Extract}(\mathsf{id}_0), \mathsf{Extract}(\mathsf{id}_0), \mathsf{Extract}(\mathsf{id}_1))$, where the third key is extracted from an independent id, for uniformly sampled id_0 and $\mathsf{id}_1 \in \mathcal{I}$. Let $(\mathsf{sk}_0, \mathsf{sk}_1, \mathsf{sk}_2)$ be the tuple the adversary receives, and for which it has to decide its form. We further expand sk_i to $(\mathsf{d}_{\mathsf{i}0}, \mathsf{d}_{\mathsf{i}1}, \mathsf{d}_{\mathsf{i}2})$ according to our scheme. If the keys were generated honestly, i.e. by following the algorithm $\mathsf{Extract}$ as described in Fig. 9, the adversary simply has to check if

$$\mathbf{e}(\frac{\mathsf{d}_{10}}{\mathsf{d}_{00}}, \frac{\mathsf{d}_{21}}{\mathsf{d}_{01}}) \stackrel{?}{=} \mathbf{e}(\frac{\mathsf{d}_{00}}{\mathsf{d}_{20}}, \frac{\mathsf{d}_{01}}{\mathsf{d}_{11}})$$

to determine the form of the tuple with overwhelming probability. If the result from the equality is true, then the three secret keys are very likely to have been extracted for the same id[3]. If the result is false, then the tuple is definitely of the form $(\mathsf{Extract}(\mathsf{id}_0), \mathsf{Extract}(\mathsf{id}_0), \mathsf{Extract}(\mathsf{id}_1))$. For completeness, we show this by expanding and simplifying the above expression.

$$\begin{aligned}
\mathbf{e}(\frac{\mathsf{d}_{10}}{\mathsf{d}_{00}}, \frac{\mathsf{d}_{21}}{\mathsf{d}_{01}}) &= \mathbf{e}(\frac{\mathsf{d}_{00}}{\mathsf{d}_{20}}, \frac{\mathsf{d}_{01}}{\mathsf{d}_{11}}) \Leftrightarrow \\
\mathbf{e}(\frac{\mathsf{g}^{\mathsf{r}_1\mathsf{t}_1\mathsf{t}_2}}{\mathsf{g}^{\mathsf{r}_0\mathsf{t}_1\mathsf{t}_2}}, \frac{\mathsf{g}^{-\mathsf{w}\mathsf{t}_2} \cdot \mathsf{h}_2^{-\mathsf{r}_2\mathsf{t}_2}}{\mathsf{g}^{-\mathsf{w}\mathsf{t}_2} \cdot \mathsf{h}_0^{-\mathsf{r}_0\mathsf{t}_2}}) &= \mathbf{e}(\frac{\mathsf{g}^{\mathsf{r}_0\mathsf{t}_1\mathsf{t}_2}}{\mathsf{g}^{\mathsf{r}_2\mathsf{t}_1\mathsf{t}_2}}, \frac{\mathsf{g}^{-\mathsf{w}\mathsf{t}_2} \cdot \mathsf{h}_0^{-\mathsf{r}_0\mathsf{t}_2}}{\mathsf{g}^{-\mathsf{w}\mathsf{t}_2} \cdot \mathsf{h}_1^{-\mathsf{r}_1\mathsf{t}_2}}) \Leftrightarrow \\
\mathbf{e}(\frac{\mathsf{g}^{\mathsf{r}_1\mathsf{t}_1\mathsf{t}_2}}{\mathsf{g}^{\mathsf{r}_0\mathsf{t}_1\mathsf{t}_2}}, \frac{\mathsf{h}_2^{-\mathsf{r}_2\mathsf{t}_2}}{\mathsf{h}_0^{-\mathsf{r}_0\mathsf{t}_2}}) &= \mathbf{e}(\frac{\mathsf{g}^{\mathsf{r}_0\mathsf{t}_1\mathsf{t}_2}}{\mathsf{g}^{\mathsf{r}_2\mathsf{t}_1\mathsf{t}_2}}, \frac{\mathsf{h}_0^{-\mathsf{r}_0\mathsf{t}_2}}{\mathsf{h}_0^{-\mathsf{r}_1\mathsf{t}_2}}) \Leftrightarrow \\
\mathbf{e}(\mathsf{g}^{(\mathsf{r}_1-\mathsf{r}_0)}, \mathsf{h}_0^{\mathsf{r}_0} \cdot \mathsf{h}_2^{-\mathsf{r}_2})^{\mathsf{t}_1(\mathsf{t}_2)^2} &= \mathbf{e}(\mathsf{g}^{(\mathsf{r}_0-\mathsf{r}_2)}, \mathsf{h}_0^{(\mathsf{r}_1-\mathsf{r}_0)})^{\mathsf{t}_1(\mathsf{t}_2)^2} \Leftrightarrow \\
\mathbf{e}(\mathsf{g}, \mathsf{h}_0^{\mathsf{r}_0} \cdot \mathsf{h}_2^{-\mathsf{r}_2}) &= \mathbf{e}(\mathsf{g}, \mathsf{h}_0^{(\mathsf{r}_0-\mathsf{r}_2)}) \Leftrightarrow \\
\mathsf{h}_2 &= \mathsf{h}_0
\end{aligned}$$

[3] Collisions in the hash function H may lead to misleading results but only occur with negligible probability.

It is now clear that IBE scheme Π [Fig. 9] fails to achieve the Strong Key Unlinkability property.

A SCHEME WITH STRONG KEY UNLINKABILITY. We extend Π to groups of composite order and obtain Π' [Fig. 10]. The extension is very simple: let all the parameters in the original scheme be from the subgroup $\mathbb{G}_p$ (generated by g_p) and randomize each element of the extracted secret key by a random element from the subgroup $\mathbb{G}_q$ (generated by g_q). Note that the message space is $\mathbb{G}_T$.

$\underline{\mathsf{Setup}(1^\lambda)}$:
$(p, q, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow_\$ \mathcal{G}_\mathcal{C}(\lambda)$
$n \leftarrow pq$; $g_p \leftarrow g^q$; $g_q \leftarrow g^p$
$\Gamma \leftarrow (n, \mathbb{G}, \mathbb{G}_T, e, g, g_p, g_q)$
$w, t_1, t_2 \leftarrow_\$ \mathbb{Z}_n$
$\Omega \leftarrow e(g_p, g_p)^{t_1 t_2 w}$
$v_1 \leftarrow g_p^{t_1}$
$v_2 \leftarrow g_p^{t_2}$
$H \leftarrow_\$ \mathcal{H} : \mathcal{I} \to \mathbb{G}_p$
$pp \leftarrow (\Gamma, \Omega, v_1, v_2, H)$
$msk \leftarrow (w, t_1, t_2)$
return (msk, pp)

$\underline{\mathsf{Extract}(pp, msk, id)}$:
$(w, t_1, t_2) \leftarrow msk$
$(\Gamma, \Omega, v_1, v_2, H) \leftarrow pp$
$(n, \mathbb{G}, \mathbb{G}_T, e, g, g_p, g_q) \leftarrow \Gamma$
$r \leftarrow_\$ \mathbb{Z}_n$
$x_0, x_1, x_2 \leftarrow_\$ \mathbb{G}_q$
$h \leftarrow H(id)$
$d_0 \leftarrow x_0 \cdot g_p^{r t_1 t_2}$
$d_1 \leftarrow x_1 \cdot g_p^{-w t_2} \cdot h^{-r t_2}$
$d_2 \leftarrow x_2 \cdot g_p^{-w t_1} \cdot h^{-r t_1}$
$sk \leftarrow (d_0, d_1, d_2)$
return sk

$\underline{\mathsf{Enc}(pp, m, id)}$:
$(\Gamma, \Omega, v_1, v_2, H) \leftarrow pp$
$(n, \mathbb{G}, \mathbb{G}_T, e, g, g_p, g_q) \leftarrow \Gamma$
$s, s_1 \leftarrow_\$ \mathbb{Z}_n$; $h \leftarrow H(id)$
$\hat{c} \leftarrow \Omega^s m$; $c_0 \leftarrow h^s$; $c_1 \leftarrow v_1^{s-s_1}$; $c_2 \leftarrow v_2^{s_1}$
$c \leftarrow (\hat{c}, c_0, c_1, c_2)$; return c

$\underline{\mathsf{Dec}(pp, c, id, sk_{id})}$:
$(\Gamma, \Omega, v_1, v_2, H) \leftarrow pp$
$(n, \mathbb{G}, \mathbb{G}_T, e, g, g_p, g_q) \leftarrow \Gamma$
$(d_0, d_1, d_2) \leftarrow sk_{id}$; $(\hat{c}, c_0, c_1, c_2) \leftarrow c$
$e_0 \leftarrow e(c_0, d_0)$; $e_1 \leftarrow e(c_1, d_1)$
$e_2 \leftarrow e(c_2, d_2)$; $m \leftarrow \hat{c} \cdot e_0 \cdot e_1 \cdot e_2$
return m

Fig. 10. Anonymous IBE scheme Π' with Strong Key Unlinkability

The decryption algorithm remains correct, since

$$e_0 = e(h^s, x_0 \cdot g_p^{r t_1 t_2}) = e(h^s, g_p^{r t_1 t_2})$$
$$e_1 = e(v_1^{s-s_1}, x_1 \cdot g_p^{-w t_2} \cdot h^{-r t_2}) = e(v_1^{s-s_1}, g_p^{-w t_2} \cdot h^{-r t_2})$$
$$e_2 = e(v_2^{s_1}, x_2 \cdot g_p^{-w t_1} \cdot h^{-r t_1}) = e(v_2^{s_1}, g_p^{-w t_1} \cdot h^{-r t_1})$$

Also, *semantic security* and *anonymity* properties are not affected, assuming DBDH and DLIN hold in $\mathbb{G}_p$. We only need to prove that Π' possesses the *Strong Key Unlinkability* property.

Theorem 4 (Appendix B). *IBE scheme Π' [Fig. 10] has Strong Key Unlinkability [Definition 8], assuming CDDH is intractable [Definition 5].*

5 Conclusions and Future Directions

Our work shows that two distinct scenarios have to be considered to model trapdoor privacy: one in the presence of ciphertexts that match trapdoors, and the other in the absence of such ciphertexts. The notion of Strong Search Pattern Privacy we introduced here addresses privacy concerns up to the point where ciphertexts matching the issued trapdoors become available, after which, search patterns can no longer be hidden from an attacker. Previous models provide limited privacy guarantees against search patterns. Of theoretical interest, it remains an open problem to prove if our scheme Π' [Fig. 10] (or any other) can achieve security according to the generalized definition of *Adaptive* Key

Unlinkability. The overarching goal would be to construct an Anonymous IBE scheme which satisfies both Adaptive Key Unlinkability and Enhanced Function Privacy, simultaneously.

Acknowledgements. The present project is supported by the National Research Fund, Luxembourg.

References

1. Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P., Shi, H.: Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions. Journal of Cryptology 21(3), 350–391 (2008)
2. Bellare, M., Boldyreva, A., Micali, S.: Public-key encryption in a multi-user setting: Security proofs and improvements. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 259–274. Springer, Heidelberg (2000)
3. Bellare, M., Boldyreva, A., O'Neill, A.: Deterministic and efficiently searchable encryption. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 535–552. Springer, Heidelberg (2007)
4. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer, Heidelberg (2006)
5. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
6. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004)
7. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
8. Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-dnf formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005)
9. Boneh, D., Raghunathan, A., Segev, G.: Function-private identity-based encryption: Hiding the function in functional encryption. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 461–478. Springer, Heidelberg (2013)
10. Boneh, D., Raghunathan, A., Segev, G.: Function-private subspace-membership encryption and its applications. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 255–275. Springer, Heidelberg (2013)
11. Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 535–554. Springer, Heidelberg (2007)
12. Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (Without random oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006)
13. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (2008)

14. Nishioka, M.: Perfect keyword privacy in PEKS systems. In: Takagi, T., Wang, G., Qin, Z., Jiang, S., Yu, Y. (eds.) ProvSec 2012. LNCS, vol. 7496, pp. 175–192. Springer, Heidelberg (2012)
15. Shen, E., Shi, E., Waters, B.: Predicate privacy in encryption systems. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 457–473. Springer, Heidelberg (2009)
16. Tang, Q.: Theory and Practice of Cryptography Solutions for Secure Information Systems. In: Search in Encrypted Data: Theoretical Models and Practical Applications, pp. 84–108. IGI (2013)

A Proof of Theorem 3

Let $\mathcal{A}$ be any legitimate PPT adversary in game $\mathsf{WEAK\text{-}KEY\text{-}UNLINK}_{\Pi,\mathcal{A}}$ [Fig. 5]. By building a simulator $\mathcal{S}_2$ [Fig. 11] that plays game $\mathsf{DLIN}_{\Gamma,\mathcal{S}_2}$ [Fig. 2] and simulates game $\mathsf{WEAK\text{-}KEY\text{-}UNLINK}_{\Pi,\mathcal{A}}$ in such a way that $\mathcal{A}$'s guess can be forward to game $\mathsf{DLIN}_{\Gamma,\mathcal{S}_2}$, we upper-bound the adversary's advantage to the hardness of deciding on an instance of this problem.

The master secret key is set as following: $\mathsf{t}_1 = \mathsf{z}_1$, $\mathsf{t}_2 = \mathsf{z}_1 \cdot \mathsf{a}$ for random $\mathsf{a} \in \mathbb{Z}_\mathsf{p}$, and $\mathsf{w} = \frac{\mathsf{z}_3 \cdot \mathsf{b}}{\mathsf{z}_1}$ for random $\mathsf{b} \in \mathbb{Z}_\mathsf{p}$. Although the values of t_1, t_2 and w are unknown to $\mathcal{S}_2$, the corresponding public parameters can still be consistently computed:

$$\Omega = \mathbf{e}(\mathsf{g},\mathsf{g})^{\mathsf{t}_1\mathsf{t}_2\mathsf{w}} = \mathbf{e}(\mathsf{g},\mathsf{g})^{\mathsf{z}_1\mathsf{z}_1\mathsf{a}\frac{\mathsf{z}_3\cdot\mathsf{b}}{\mathsf{z}_1}} = \mathbf{e}(\mathsf{Z}_{13},\mathsf{g})^{\mathsf{ab}}$$
$$\mathsf{v}_1 = \mathsf{g}^{\mathsf{t}_1} = \mathsf{Z}_1$$
$$\mathsf{v}_2 = \mathsf{g}^{\mathsf{t}_2} = (\mathsf{Z}_1)^{\mathsf{a}}$$

The hash function H is modeled as a random oracle and set to $(\mathsf{g}^{\mathsf{z}_1})^{\mathsf{x}} \cdot \mathsf{g}^{-\frac{1}{\mathsf{y}}}$, for random $\mathsf{x},\mathsf{y} \in \mathbb{Z}_\mathsf{p}^2$. We assume, without loss of generality, that $\mathcal{A}$ always asks for the hash value of id before querying id to oracle Extract. Whenever asked to extract a private key on some id, we set $\mathsf{r} = \mathsf{w} \cdot \mathsf{y}$, where y is the value used to compute the hash of that particular id. Note that this still makes r uniformly distributed over $\mathbb{Z}_\mathsf{p}$ and independent of h and w. Given this, private keys can be extracted as follows:

$$\mathsf{d}_0 = \mathsf{g}^{\mathsf{r}\mathsf{t}_1\mathsf{t}_2} = \mathsf{g}^{\mathsf{w}\mathsf{y}\mathsf{t}_1\mathsf{t}_2} = \mathsf{g}^{\frac{\mathsf{z}_3\cdot\mathsf{b}}{\mathsf{z}_1}\mathsf{y}\mathsf{z}_1\mathsf{z}_1\mathsf{a}} = (\mathsf{Z}_{13})^{\mathsf{aby}}$$
$$\mathsf{d}_1 = \mathsf{g}^{-\mathsf{w}\mathsf{t}_2} \cdot \mathsf{h}^{-\mathsf{r}\mathsf{t}_2} = \mathsf{g}^{-\mathsf{w}\mathsf{t}_2} \cdot \left[(\mathsf{g}^{\mathsf{z}_1})^{\mathsf{x}} \cdot \mathsf{g}^{-\frac{1}{\mathsf{y}}}\right]^{-\mathsf{w}\mathsf{y}\mathsf{t}_2} = \mathsf{g}^{-\mathsf{z}_1\mathsf{x}\mathsf{w}\mathsf{y}\mathsf{t}_2} = \mathsf{g}^{-\mathsf{z}_1\mathsf{x}\frac{\mathsf{z}_3\cdot\mathsf{b}}{\mathsf{z}_1}\mathsf{y}\mathsf{z}_1\mathsf{a}} = (\mathsf{Z}_{13})^{-\mathsf{abxy}}$$
$$\mathsf{d}_2 = \mathsf{g}^{-\mathsf{w}\mathsf{t}_1} \cdot \mathsf{h}^{-\mathsf{r}\mathsf{t}_1} = \mathsf{g}^{-\mathsf{w}\mathsf{t}_1} \cdot \left[(\mathsf{g}^{\mathsf{z}_1})^{\mathsf{x}} \cdot \mathsf{g}^{-\frac{1}{\mathsf{y}}}\right]^{-\mathsf{w}\mathsf{y}\mathsf{t}_1} = \mathsf{g}^{-\mathsf{z}_1\mathsf{x}\mathsf{w}\mathsf{y}\mathsf{t}_1} = \mathsf{g}^{-\mathsf{z}_1\mathsf{x}\frac{\mathsf{z}_3\cdot\mathsf{b}}{\mathsf{z}_1}\mathsf{y}\mathsf{z}_1} = (\mathsf{Z}_{13})^{-\mathsf{bxy}}$$

Finally, to complete the simulation, we extract two private keys to challenge $\mathcal{A}$, such that these private keys are for the same id if $\mathcal{S}_2$ received a valid DLIN tuple, and for different ids otherwise. Let $\mathsf{sk}^\star = (\mathsf{d}_0^\star, \mathsf{d}_1^\star, \mathsf{d}_2^\star)$ and $\mathsf{sk}^\circ = (\mathsf{d}_0^\circ, \mathsf{d}_1^\circ, \mathsf{d}_2^\circ)$ be the challenge keys. We set $\mathsf{h} = \mathsf{g}^{\mathsf{z}_1\mathsf{z}_4}$, $\mathsf{r}^\star = \frac{\mathsf{b}}{(\mathsf{z}_1)^2}$ and $\mathsf{r}^\circ = \frac{\mathsf{z}_2+\mathsf{b}}{(\mathsf{z}_1)^2}$. Note that h is uniformly distributed over $\mathbb{G}$, and $\mathsf{r}^\star$ and r° are uniformly distributed over $\mathbb{Z}_\mathsf{p}$, independent of each other and of w. For completeness, we present the equalities between the original expressions and those computed by the simulator.

$$\mathsf{d}_0^\star = \mathsf{g}^{\mathsf{r}^\star \mathsf{t}_1 \mathsf{t}_2} = \mathsf{g}^{\frac{\mathsf{b}}{(\mathsf{z}_1)^2}\mathsf{z}_1\mathsf{z}_1\mathsf{a}} = \mathsf{g}^{\mathsf{ab}}$$
$$\mathsf{d}_1^\star = \mathsf{g}^{-\mathsf{wt}_2} \cdot \mathsf{h}^{-\mathsf{r}^\star \mathsf{t}_2} = \mathsf{g}^{-\frac{\mathsf{z}_3\mathsf{b}}{\mathsf{z}_1}\mathsf{z}_1\mathsf{a}} \cdot \left(\mathsf{g}^{\mathsf{z}_1\mathsf{z}_4}\right)^{-\frac{\mathsf{b}}{(\mathsf{z}_1)^2}\mathsf{z}_1\mathsf{a}} = \left(\mathsf{g}^{-\mathsf{ab}}\right)^{\mathsf{z}_3} \cdot \left(\mathsf{g}^{-\mathsf{ab}}\right)^{\mathsf{z}_4} = \mathsf{Z}^{-\mathsf{ab}}$$
$$\mathsf{d}_1^\star = \mathsf{g}^{-\mathsf{wt}_1} \cdot \mathsf{h}^{-\mathsf{r}^\star \mathsf{t}_1} = \mathsf{g}^{-\frac{\mathsf{z}_3\mathsf{b}}{\mathsf{z}_1}\mathsf{z}_1} \cdot \left(\mathsf{g}^{\mathsf{z}_1\mathsf{z}_4}\right)^{-\frac{\mathsf{b}}{(\mathsf{z}_1)^2}\mathsf{z}_1} = \left(\mathsf{g}^{-\mathsf{b}}\right)^{\mathsf{z}_3} \cdot \left(\mathsf{g}^{-\mathsf{b}}\right)^{\mathsf{z}_4} = \mathsf{Z}^{-\mathsf{b}}$$
$$\mathsf{d}_0^\circ = \mathsf{g}^{\mathsf{r}^\circ \mathsf{t}_1 \mathsf{t}_2} = \mathsf{g}^{\frac{\mathsf{z}_2+\mathsf{b}}{(\mathsf{z}_1)^2}\mathsf{z}_1\mathsf{z}_1\cdot\mathsf{a}} = \mathsf{g}^{\mathsf{z}_2\cdot\mathsf{a}+\mathsf{ab}} = (\mathsf{Z}_2)^{\mathsf{a}} \cdot \mathsf{g}^{\mathsf{ab}}$$
$$\mathsf{d}_1^\circ = \mathsf{g}^{-\mathsf{wt}_2} \cdot \mathsf{h}^{-\mathsf{r}^\circ \mathsf{t}_2} = \mathsf{g}^{-\frac{\mathsf{z}_3\mathsf{b}}{\mathsf{z}_1}\mathsf{z}_1\mathsf{a}} \cdot \left(\mathsf{g}^{\mathsf{z}_1\mathsf{z}_4}\right)^{-\frac{\mathsf{z}_2+\mathsf{b}}{(\mathsf{z}_1)^2}\mathsf{z}_1\mathsf{a}} = \left(\mathsf{g}^{-\mathsf{ab}}\right)^{(\mathsf{z}_3+\mathsf{z}_4)} \cdot \left(\mathsf{g}^{\mathsf{z}_2\mathsf{z}_4}\right)^{-\mathsf{a}} = \mathsf{Z}^{-\mathsf{ab}} \cdot (\mathsf{Z}_{24})^{-\mathsf{a}}$$
$$\mathsf{d}_2^\circ = \mathsf{g}^{-\mathsf{wt}_1} \cdot \mathsf{h}^{-\mathsf{r}^\circ \mathsf{t}_1} = \mathsf{g}^{-\frac{\mathsf{z}_3\mathsf{b}}{\mathsf{z}_1}\mathsf{z}_1} \cdot \left(\mathsf{g}^{\mathsf{z}_1\mathsf{z}_4}\right)^{-\frac{\mathsf{z}_2+\mathsf{b}}{(\mathsf{z}_1)^2}\mathsf{z}_1} = \left(\mathsf{g}^{-\mathsf{b}}\right)^{(\mathsf{z}_3+\mathsf{z}_4)} \cdot \left(\mathsf{g}^{\mathsf{z}_2\mathsf{z}_4}\right)^{-1} = \mathsf{Z}^{-\mathsf{b}} \cdot (\mathsf{Z}_{24})^{-1}$$

Therefore, we have that $\mathbf{Adv}^{\mathsf{WEAK\text{-}KEY\text{-}UNLINK}}_{\Pi,\mathcal{A}}(\lambda) = \mathbf{Adv}^{\mathsf{DLIN}}_{\Gamma,\mathcal{S}_2}$, which concludes our proof. □

procedure Initialize(λ):
$(\mathsf{Z}_1, \mathsf{Z}_2, \mathsf{Z}_{13}, \mathsf{Z}_{24}, \mathsf{Z}) \leftarrow \mathsf{DLIN.Initialize}$
$\mathsf{a} \leftarrow_\$ \mathbb{Z}_\mathsf{p}, \mathsf{b} \leftarrow_\$ \mathbb{Z}_\mathsf{p}$
$\mathsf{list}_\mathsf{H} \leftarrow [\,]$

$\Omega \leftarrow \mathbf{e}(\mathsf{Z}_{13}, \mathsf{g})^{\mathsf{ab}}$
$\mathsf{v}_1 \leftarrow \mathsf{Z}_1$
$\mathsf{v}_2 \leftarrow (\mathsf{Z}_1)^{\mathsf{a}}$

$\mathsf{d}_0^\star \leftarrow \mathsf{g}^{\mathsf{ab}}$, $\mathsf{d}_0^\circ \leftarrow (\mathsf{Z}_2)^{\mathsf{a}} \cdot \mathsf{g}^{\mathsf{ab}}$
$\mathsf{d}_1^\star \leftarrow \mathsf{Z}^{-\mathsf{ab}}$, $\mathsf{d}_1^\circ \leftarrow \mathsf{Z}^{-\mathsf{ab}} \cdot (\mathsf{Z}_{24})^{-\mathsf{a}}$
$\mathsf{d}_1^\star \leftarrow \mathsf{Z}^{-\mathsf{b}}$, $\mathsf{d}_2^\circ \leftarrow \mathsf{Z}^{-\mathsf{b}} \cdot (\mathsf{Z}_{24})^{-1}$

$\mathsf{sk}_0 \leftarrow (\mathsf{d}_0^\star, \mathsf{d}_2^\star, \mathsf{d}_2^\star)$
$\mathsf{sk}_1 \leftarrow (\mathsf{d}_0^\circ, \mathsf{d}_2^\circ, \mathsf{d}_2^\circ)$
$\mathsf{pp} \leftarrow (\Omega, \mathsf{v}_1, \mathsf{v}_2)$

return $(\mathsf{pp}, \mathsf{sk}_0, \mathsf{sk}_1)$

procedure H(id) :
get (x, y) for id from list_H
if (x, y) $==\perp$
..... $\mathsf{x} \leftarrow_\$ \mathbb{Z}_\mathsf{p}$
..... $\mathsf{y} \leftarrow_\$ \mathbb{Z}_\mathsf{p}$
..... $\mathsf{list}_\mathsf{H} \leftarrow (\mathsf{id}, \mathsf{x}, \mathsf{y}) : \mathsf{list}_\mathsf{H}$
$\mathsf{h} \leftarrow (\mathsf{g}^{\mathsf{z}_1})^{\mathsf{x}} \cdot \mathsf{g}^{-\frac{1}{\mathsf{y}}}$
return h

procedure Extract(id):
get (x, y) for id from list_H

$\mathsf{d}_0 \leftarrow (\mathsf{Z}_{13})^{\mathsf{aby}}$
$\mathsf{d}_1 \leftarrow (\mathsf{Z}_{13})^{-\mathsf{abxy}}$
$\mathsf{d}_2 \leftarrow (\mathsf{Z}_{13})^{-\mathsf{bxy}}$

$\mathsf{sk}_{\mathsf{id}} \leftarrow (\mathsf{d}_0, \mathsf{d}_1, \mathsf{d}_2)$
return $\mathsf{sk}_{\mathsf{id}}$

procedure Finalize(bit):
DLIN.Finalize(bit)

Fig. 11. Simulator $\mathcal{S}_2$ forwards $\mathcal{A}$'s guess from game $\mathsf{WEAK\text{-}KEY\text{-}UNLINK}_{\Pi,\mathcal{A}}$ to game $\mathsf{DLIN}_{\Gamma,\mathcal{S}_2}$

B Proof of Theorem 4

First, let us show an important re-randomization property that scheme Π' possess and that is relevant for the completion of this proof. From two keys honestly extracted from the same identity, say $\mathsf{sk}_0 = (\mathsf{d}_{00}, \mathsf{d}_{01}, \mathsf{d}_{02})$ and $\mathsf{sk}_1 = (\mathsf{d}_{10}, \mathsf{d}_{11}, \mathsf{d}_{12})$, one can generate new valid keys for that identity with fresh random coins, without the knowledge of any secret parameter. Concretely, $\mathsf{sk}_2 = (\mathsf{d}_{20}, \mathsf{d}_{21}, \mathsf{d}_{22})$ can be generated as follows, with a random $\mathsf{y} \in \mathbb{Z}_\mathsf{n}$ and random $\mathsf{R}_0, \mathsf{R}_1, \mathsf{R}_2 \in \mathbb{G}_\mathsf{q}$:

$$\mathsf{d}_{20} = \mathsf{R}_0 \cdot \left(\tfrac{\mathsf{d}_{10}}{\mathsf{d}_{00}}\right)^{\mathsf{y}} \cdot \mathsf{d}_{00} = \left[\mathsf{R}_0 \cdot \tfrac{(\mathsf{x}_{10})^{\mathsf{y}}}{(\mathsf{x}_{00})^{(\mathsf{y}-1)}}\right] \cdot \mathsf{g}^{[\mathsf{y}\mathsf{r}_1-(\mathsf{y}-1)\mathsf{r}_0]\mathsf{t}_1\mathsf{t}_2}$$
$$\mathsf{d}_{21} = \mathsf{R}_1 \cdot \left(\tfrac{\mathsf{d}_{11}}{\mathsf{d}_{01}}\right)^{\mathsf{y}} \cdot \mathsf{d}_{01} = \left[\mathsf{R}_1 \cdot \tfrac{(\mathsf{x}_{11})^{\mathsf{y}}}{(\mathsf{x}_{01})^{(\mathsf{y}-1)}}\right] \cdot \mathsf{g}^{-\mathsf{wt}_2} \cdot \mathsf{h}^{-[\mathsf{y}\mathsf{r}_1-(\mathsf{y}-1)\mathsf{r}_0]\mathsf{t}_2}$$
$$\mathsf{d}_{22} = \mathsf{R}_2 \cdot \left(\tfrac{\mathsf{d}_{12}}{\mathsf{d}_{02}}\right)^{\mathsf{y}} \cdot \mathsf{d}_{02} = \left[\mathsf{R}_2 \cdot \tfrac{(\mathsf{x}_{12})^{\mathsf{y}}}{(\mathsf{x}_{02})^{(\mathsf{y}-1)}}\right] \cdot \mathsf{g}^{-\mathsf{wt}_1} \cdot \mathsf{h}^{-[\mathsf{y}\mathsf{r}_1-(\mathsf{y}-1)\mathsf{r}_0]\mathsf{t}_1}$$[1.5mm]

Let $\mathcal{A}$ be any PPT adversary against $\mathsf{STRONG\text{-}KEY\text{-}UNLINK}_{\Pi',\mathcal{A}}$ [Fig. 6]. We now drastically simplify the security model, so that it looks like the one presented

in Fig. 12, which we call 5-KEY-UNLINK. Using a hybrid argument and taking advantage of the re-randomization property previously described, we show that the advantage of $\mathcal{A}$ against STRONG-KEY-UNLINK$_{\Pi',\mathcal{A}}$ is polynomially-bounded by the advantage of $\mathcal{A}$ against 5-KEY-UNLINK.

procedure Initialize(λ):
$(\mathsf{msk}, \mathsf{pp}) \leftarrow_{\$} \mathsf{Setup}(1^{\lambda})$
$\mathsf{bit} \leftarrow_{\$} \{0,1\}$
$\mathsf{id}_0 \leftarrow_{\$} \mathcal{I}$
$\mathsf{id}_1 \leftarrow_{\$} \mathcal{I}$
$\mathsf{sk}_0 \leftarrow_{\$} \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}_0)$
$\mathsf{sk}_1 \leftarrow_{\$} \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}_0)$
$\mathsf{sk}_2 \leftarrow_{\$} \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}_{\mathsf{bit}})$
$\mathsf{sk}_3 \leftarrow_{\$} \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}_1)$
$\mathsf{sk}_4 \leftarrow_{\$} \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}_1)$
return $(\mathsf{pp}, \mathsf{sk}_0, \mathsf{sk}_1, \mathsf{sk}_2, \mathsf{sk}_3, \mathsf{sk}_4)$

procedure Extract(id):
$\mathsf{sk}_{\mathsf{id}} \leftarrow_{\$} \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id})$
return $\mathsf{sk}_{\mathsf{id}}$

procedure Finalize(bit′):
return $(\mathsf{bit} = \mathsf{bit}')$

Fig. 12. 5-KEY-UNLINK$_{\Pi,\mathcal{A}}$ Game

In STRONG-KEY-UNLINK$_{\Pi',\mathcal{A}}$, $\mathcal{A}$ submits two lists list_0 and list_1 of the same length, say L, for the challenge. For this argument, we construct L + 1 lists. The first list is list_0 and the last list is list_1. In between, we have L-1 intermediate lists that transition from list_0 to list_1, one element at the time. The L-1 intermediate lists are constructed such that the first list is list_0, and for every $\mathsf{i} \in \{1..\mathsf{L}\text{-}1\}$, $\mathsf{list}^{\mathsf{i}} = \mathsf{list}^{\mathsf{i}-1}$, except for the element $\mathsf{list}^{\mathsf{i}}[\mathsf{i}]$ which is taken from $\mathsf{list}_1[\mathsf{i}]$. Again, the last list is list_1. The advantage $\mathcal{A}$ has in distinguishing list_0 from list_1 cannot be more than the sum of the advantages of distinguishing $\mathsf{list}^{\mathsf{i}-1}$ from $\mathsf{list}^{\mathsf{i}}$, for every $\mathsf{i} \in \{1..(\mathsf{L}+1)\}$. The probability of distinguishing $\mathsf{list}^{\mathsf{i}-1}$ from $\mathsf{list}^{\mathsf{i}}$ cannot be more than that of identifying the form of the tuple in model 5-KEY-UNLINK. More precisely, one can expand the 5-tuple $(\mathsf{sk}_0^\circ, \mathsf{sk}_1^\circ, \mathsf{sk}_2^\circ, \mathsf{sk}_3^\circ, \mathsf{sk}_4^\circ)$ from 5-KEY-UNLINK into a L-tuple of keys that corresponds to the requirements of either $\mathsf{list}^{\mathsf{i}-1}$ or $\mathsf{list}^{\mathsf{i}}$. Since the lists only (possibly) differ in position i, we set sk_{i} of the L-tuple to sk_2°. Every other key is extracted from the extraction oracle of model 5-KEY-UNLINK or generated from $(\mathsf{sk}_0^\circ, \mathsf{sk}_1^\circ)$ or $(\mathsf{sk}_3^\circ, \mathsf{sk}_4^\circ)$ if the key is required to be extracted from the identity in $\mathsf{list}^{\mathsf{i}-1}[\mathsf{i}]$ or $\mathsf{list}^{\mathsf{i}}[\mathsf{i}]$, respectively.

The model can be further simplified to that of Fig. 13, which we call 4-KEY-UNLINK. Again, we make use of the so-called hybrid argument and the re-randomization property introduced in the beginning of this proof that Π' possesses[4], the difficulty of distinguishing a 5-tuple of keys extracted from $(\mathsf{id}_0, \mathsf{id}_0, \mathsf{id}_0, \mathsf{id}_1, \mathsf{id}_1)$ from those extracted from $(\mathsf{id}_0, \mathsf{id}_0, \mathsf{id}_0, \mathsf{id}_0, \mathsf{id}_0)$, where id_0 and id_1 are sampled from $\mathcal{I}$, is equivalent to that of distinguishing a 4-tuple of keys that were extracted from $(\mathsf{id}_0, \mathsf{id}_0, \mathsf{id}_1, \mathsf{id}_1)$ from those extracted from $(\mathsf{id}_0, \mathsf{id}_0, \mathsf{id}_0, \mathsf{id}_0)$, since the fifth key the adversary could generate himself. This difficulty of distinguishing the 5-tuple of keys extracted from $(\mathsf{id}_0, \mathsf{id}_0, \mathsf{id}_1, \mathsf{id}_1, \mathsf{id}_1)$ from those extracted from $(\mathsf{id}_0, \mathsf{id}_0, \mathsf{id}_0, \mathsf{id}_0, \mathsf{id}_0)$ is also the same as distinguishing the key tuple in 4-KEY-UNLINK model. So, the advantage $\mathcal{A}$ has in distinguishing the

[4] From two keys honestly extracted for the same identity, we can generate a third one with random coins.

tuples in 5-KEY-UNLINK game cannot be more than *twice* the advantage $\mathcal{A}$ has in distinguishing the tuples in 4-KEY-UNLINK.

procedure Initialize(λ):
$(\mathsf{msk}, \mathsf{pp}) \leftarrow_\$ \mathsf{Setup}(1^\lambda)$
$\mathsf{bit} \leftarrow_\$ \{0,1\}$
$\mathsf{id}_0 \leftarrow_\$ \mathcal{I}$
$\mathsf{id}_1 \leftarrow_\$ \mathcal{I}$
$\mathsf{sk}_0 \leftarrow_\$ \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}_0)$
$\mathsf{sk}_1 \leftarrow_\$ \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}_0)$
$\mathsf{sk}_2 \leftarrow_\$ \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}_{\mathsf{bit}})$
$\mathsf{sk}_3 \leftarrow_\$ \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id}_{\mathsf{bit}})$
return $(\mathsf{pp}, \mathsf{sk}_0, \mathsf{sk}_1, \mathsf{sk}_2, \mathsf{sk}_3)$

procedure Extract(id):
$\mathsf{sk}_{\mathsf{id}} \leftarrow_\$ \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id})$
return $\mathsf{sk}_{\mathsf{id}}$

procedure Finalize(bit'):
return $(\mathsf{bit} = \mathsf{bit}')$

Fig. 13. 4-KEY-UNLINK$_{\Pi,\mathcal{A}}$ Game

procedure Initialize(λ):
$(\Gamma, \mathsf{Z_a}, \mathsf{Z_b}, \mathsf{Z_{ab}}) \leftarrow \mathsf{CDDH.Initialize}(\lambda)$
$(\mathsf{n}, \mathbb{G}, \mathbb{G}_\mathsf{T}, \mathbf{e}, \mathsf{g}, \mathsf{g_p}, \mathsf{g_q}) \leftarrow \Gamma$
$\mathsf{w}, \mathsf{t}_1, \mathsf{t}_2 \leftarrow_\$ \mathbb{Z}_\mathsf{n}$
$\Omega \leftarrow \mathbf{e}(\mathsf{g_p}, \mathsf{g_p})^{\mathsf{t}_1 \mathsf{t}_2 \mathsf{w}}$
$\mathsf{v}_1 \leftarrow \mathsf{g_p}^{\mathsf{t}_1}$
$\mathsf{v}_2 \leftarrow \mathsf{g_p}^{\mathsf{t}_2}$
$\mathsf{msk} \leftarrow (\mathsf{w}, \mathsf{t}_1, \mathsf{t}_2)$
$\mathsf{pp} \leftarrow (\Gamma, \Omega, \mathsf{v}_1, \mathsf{v}_2)$

$\mathsf{r}_0^\star \leftarrow_\$ \mathbb{Z}_\mathsf{n}$
$\mathsf{x}_{00}', \mathsf{x}_{01}', \mathsf{x}_{02}' \leftarrow_\$ \mathbb{Z}_\mathsf{n}; \quad \mathsf{x}_{00} \leftarrow \mathsf{g_q}^{\mathsf{x}_{00}'}; \ \mathsf{x}_{01} \leftarrow \mathsf{g_q}^{\mathsf{x}_{01}'}; \ \mathsf{x}_{02} \leftarrow \mathsf{g_q}^{\mathsf{x}_{02}'}$
$\mathsf{sk}_0^\star \leftarrow (\mathsf{x}_{00} \cdot (\mathsf{g_p})^{\mathsf{r}_0^\star \mathsf{t}_1 \mathsf{t}_2}, \mathsf{x}_{01} \cdot \mathsf{Z_a}^{-\mathsf{r}_0^\star \mathsf{t}_2} \cdot (\mathsf{g_p})^{-\mathsf{w}\mathsf{t}_2}, \mathsf{x}_{02} \cdot \mathsf{Z_a}^{-\mathsf{r}_0^\star \mathsf{t}_1} \cdot (\mathsf{g_p})^{-\mathsf{w}\mathsf{t}_1})$

$\mathsf{r}_1^\star \leftarrow_\$ \mathbb{Z}_\mathsf{n}$
$\mathsf{x}_{10}', \mathsf{x}_{11}', \mathsf{x}_{12}' \leftarrow_\$ \mathbb{Z}_\mathsf{n}; \quad \mathsf{x}_{10} \leftarrow \mathsf{g_q}^{\mathsf{x}_{10}'}; \ \mathsf{x}_{11} \leftarrow \mathsf{g_q}^{\mathsf{x}_{11}'}; \ \mathsf{x}_{12} \leftarrow \mathsf{g_q}^{\mathsf{x}_{12}'}$
$\mathsf{sk}_1^\star \leftarrow (\mathsf{x}_{10} \cdot (\mathsf{g_p})^{\mathsf{r}_1^\star \mathsf{t}_1 \mathsf{t}_2}, \mathsf{x}_{11} \cdot \mathsf{Z_a}^{-\mathsf{r}_1^\star \mathsf{t}_2} \cdot (\mathsf{g_p})^{-\mathsf{w}\mathsf{t}_2}, \mathsf{x}_{12} \cdot \mathsf{Z_a}^{-\mathsf{r}_1^\star \mathsf{t}_1} \cdot (\mathsf{g_p})^{-\mathsf{w}\mathsf{t}_1})$

$\mathsf{x}_{20}', \mathsf{x}_{21}', \mathsf{x}_{22}' \leftarrow_\$ \mathbb{Z}_\mathsf{n}; \quad \mathsf{x}_{20} \leftarrow \mathsf{g_q}^{\mathsf{x}_{20}'}; \ \mathsf{x}_{21} \leftarrow \mathsf{g_q}^{\mathsf{x}_{21}'}; \ \mathsf{x}_{22} \leftarrow \mathsf{g_q}^{\mathsf{x}_{22}'}$
$\mathsf{sk}_2^\star \leftarrow (\mathsf{x}_{20} \cdot \mathsf{Z_b}^{\mathsf{t}_1 \mathsf{t}_2}, \mathsf{x}_{21} \cdot \mathsf{Z_{ab}}^{-\mathsf{t}_2} \cdot (\mathsf{g_p})^{-\mathsf{w}\mathsf{t}_2}, \mathsf{x}_{22} \cdot \mathsf{Z_{ab}}^{-\mathsf{t}_1} \cdot (\mathsf{g_p})^{-\mathsf{w}\mathsf{t}_1})$

$\mathsf{u} \leftarrow_\$ \mathbb{Z}_\mathsf{n}$
$\mathsf{x}_{30}', \mathsf{x}_{31}', \mathsf{x}_{32}' \leftarrow_\$ \mathbb{Z}_\mathsf{n}; \quad \mathsf{x}_{30} \leftarrow \mathsf{g_q}^{\mathsf{x}_{30}'}; \ \mathsf{x}_{31} \leftarrow \mathsf{g_q}^{\mathsf{x}_{31}'}; \ \mathsf{x}_{32} \leftarrow \mathsf{g_q}^{\mathsf{x}_{32}'}$
$\mathsf{sk}_3^\star \leftarrow (\mathsf{x}_{30} \cdot \mathsf{Z_b}^{\mathsf{u}\mathsf{t}_1 \mathsf{t}_2}, \mathsf{x}_{31} \cdot \mathsf{Z_{ab}}^{-\mathsf{u}\mathsf{t}_2} \cdot (\mathsf{g_p})^{-\mathsf{w}\mathsf{t}_2}, \mathsf{x}_{32} \cdot \mathsf{Z_{ab}}^{-\mathsf{u}\mathsf{t}_1} \cdot (\mathsf{g_p})^{-\mathsf{w}\mathsf{t}_1})$

return $(\mathsf{pp}, \mathsf{sk}_0^\star, \mathsf{sk}_1^\star, \mathsf{sk}_2^\star, \mathsf{sk}_3^\star)$

procedure Extract(id):
$\mathsf{sk}_{\mathsf{id}} \leftarrow_\$ \mathsf{Extract}(\mathsf{pp}, \mathsf{msk}, \mathsf{id})$
return $\mathsf{sk}_{\mathsf{id}}$

procedure Finalize(bit):
return $\mathsf{CDDH.Finalize}(\mathsf{bit})$

Fig. 14. Simulator $\mathcal{S}_3$ forwards $\mathcal{A}$'s guess from 4-KEY-ANO$_{\Pi',\mathcal{A}}$ to game CDDH

To complete the proof, we build a simulator $\mathcal{S}_3$ [Fig. 14] that by playing game CDDH$_{\Gamma',\mathcal{S}_3}$ outputs four keys $(\mathsf{sk}_0^\star, \mathsf{sk}_1^\star, \mathsf{sk}_2^\star, \mathsf{sk}_3^\star)$ such that the adversary's guess in 4-KEY-UNLINK$_{\Pi',\mathcal{A}}$ can be forward to game CDDH$_{\Gamma',\mathcal{S}_3}$. We refer to key $\mathsf{sk}_\mathsf{i}^\star$ as the tuple $(\mathsf{d}_{\mathsf{i}0}^\star, \mathsf{d}_{\mathsf{i}1}^\star, \mathsf{d}_{\mathsf{i}2}^\star)$, associated with $\mathsf{h}_\mathsf{i}^\star$, the hashed-identity from which $\mathsf{sk}_\mathsf{i}^\star$ was extracted. If the simulator receives a well-formed CDDH tuple, $\mathsf{h}_0^\star = \mathsf{h}_1^\star = \mathsf{h}_2^\star = \mathsf{h}_3^\star$ is set to g^a. Otherwise, $\mathsf{h}_0^\star = \mathsf{h}_1^\star = \mathsf{g}^\mathsf{a}$ and $\mathsf{h}_2^\star = \mathsf{h}_3^\star$ with an independent random value in $\mathbb{G}_\mathsf{p}$. We also set $\mathsf{r}_2^\star = \mathsf{b}$ and $\mathsf{r}_3^\star = \mathsf{b} \cdot \mathsf{u}$, for a random $\mathsf{u} \in \mathbb{Z}_\mathsf{n}$. Finally, we have that $\mathbf{Adv}_{\Pi',\mathcal{A}}^{\mathsf{STRONG\text{-}KEY\text{-}UNLINK}}(\lambda) \leq 2\mathsf{L} \cdot \mathbf{Adv}_{\Gamma',\mathcal{S}_3}^{\mathsf{CDDH}}$, which concludes our proof. □