Trapdoor Privacy in Asymmetric Searchable Encryption Schemes

Afonso Arriaga[1] and Qiang Tang[1]

APSIA group, SnT, University of Luxembourg
6, rue Richard Coudenhove-Kalergi, L-1359 Luxembourg
{afonso.delerue,qiang.tang}@uni.lu

Abstract. We investigate the open problem, namely trapdoor privacy, in asymmetric searchable encryption (ASE) schemes. We first present two trapdoor privacy definitions (i.e. 2-TRAP-PRIV and poly-TRAP-PRIV) which provide different levels of security guarantee. Motivated by the generic transformation from IBE to ASE, we introduce two key anonymity properties (i.e. 2-KEY-ANO and poly-KEY-ANO) for IBE schemes, so that these properties directly lead to the resulting ASE's 2-TRAP-PRIV and poly-TRAP-PRIV properties respectively at the end of a transformation. We then present a simplified Boyen-Waters scheme and prove that it achieves IBE-IND-CPA, IBE-ANO (anonymity), and 2-KEY-ANO security in the random oracle model. Finally, we extend the simplified Boyen-Waters scheme to be based on pairings over composite-order groups and prove that the extended scheme achieves poly-KEY-ANO security without random oracles.

Keywords: Searchable Encryption, Trapdoor Privacy, Anonymous IBE.

1 Introduction

A searchable encryption (SE) scheme allows a third party to search over a client's encrypted data, on its behalf, without the need of recovering the plaintexts. In a cloud computing era, SE schemes enable organizations and individuals to outsource their data in encrypted form and securely delegate the search functionality to the cloud service provider. As such, the concept of SE has received a great deal of attention and resulted in many cryptographic constructions. Mainly, existing schemes, as surveyed in [1], fall into either one of these two settings: symmetric or asymmetric.

- In the symmetric setting, which was first introduced by Song, Wagner and Perrig [2], the client encrypts her own data and stores the resulting ciphertexts in a remote server. Later on, she can authorize the server to search on her behalf by issuing a trapdoor for a target message, generated from her secret key. We refer to schemes in this setting as *symmetric* SE schemes.

- The asymmetric setting was first introduced by Boneh *et al.* [3]. In this setting, the client generates a public/private key pair and publishes her public key. With this public key, any entity can generate encrypted contents that only the client can decrypt. Unlike ciphertexts generated from standard public-key encryption schemes, these are searchable upon delegation, meaning that if stored in a remote server, the client can authorize the server to search on her behalf by issuing a trapdoor for a target message, generated from her secret key. We refer to schemes in this setting as *asymmetric* SE schemes.

In both cases, the authors from [2, 3] (and many others that followed), only considered search queries based on equality tests, where a match is determined by whether the message encoded in the trapdoor is equal to the plaintext underneath the ciphertext. Other authors, as mentioned in [1], have considered search queries with more complex

comparison structures, allowing conjunctive, disjunctive, subset, inner product and stemming types of queries. Here, we concentrate on SE schemes that support equality tests. Nevertheless, our observations apply to the other SE schemes as well.

1.1 Privacy Issues in SE Schemes

The search functionality itself leaks inevitably *some* information to the (semi-trusted) server, which is the primary attacker for SE schemes. However, this leakage should not reveal more than that revealed by the matches between the issued trapdoors and the ciphertexts (and other information that one could infer from such matches, e.g. when two ciphertexts are matched by the same trapdoor it generally means that the two ciphertexts encrypt the same message). Thus, the leakage of information can occur from two interdependent sources, namely the ciphertexts and the trapdoors. The interdependency lies in the fact that information leakage from one source will lead to an information leakage of the other. For example, if an attacker can recover some information from a ciphertext (e.g. the plaintext is not a particular message m) then it can deduce some information from a given trapdoor that matches the ciphertext (i.e. the message encoded in the trapdoor is not m), and the reverse also applies. As a result, in order to capture the *ideal* security for SE schemes, it is essential to take into account the information leakage from both ciphertexts and trapdoors simultaneously.

For symmetric SE schemes, Shen, Shi and Waters [4] proposed a model (i.e. *Full Security*) to capture information leakages from both ciphertexts and trapdoors[1]. In their model, the challenger randomly picks a bit bit and answers a polynomial number of encryption and trapdoor queries: for a encryption oracle query with $\{\mathsf{m}_{\mathsf{i}_0}, \mathsf{m}_{\mathsf{i}_1}\}$ the challenger returns the ciphertext for $\mathsf{m}_{\mathsf{i}_{\mathsf{bit}}}$, and for a trapdoor oracle query with $\{\mathsf{m}_{\mathsf{j}_0}, \mathsf{m}_{\mathsf{j}_1}\}$ the challenger returns the trapdoor for $\mathsf{m}_{\mathsf{j}_{\mathsf{bit}}}$. The attacker issues the oracle queries adaptively under the restriction that $\mathsf{m}_{\mathsf{i}_0} = \mathsf{m}_{\mathsf{j}_0}$ iff $\mathsf{m}_{\mathsf{i}_1} = \mathsf{m}_{\mathsf{j}_1}$, and outputs a guess bit' for bit. If a symmetric SE scheme is secure (i.e. $|\Pr[\mathsf{bit}' = \mathsf{bit}] - \frac{1}{2}|$ is negligible for any attacker), then it only leaks the information "which ciphertext matches which trapdoor".

Unfortunately, for asymmetric SE schemes, no security model has simultaneously captured information leakages from both ciphertexts and trapdoors. Most existing security models are in the vein of that from [3] and have only considered information leakage from ciphertexts. Informally, these security models guarantee that, with a provably secure SE scheme, if an attacker has not been assigned the trapdoors for two messages $\{\mathsf{m}_0, \mathsf{m}_1\}$ then it is not able to distinguish their ciphertexts. Inherently, these models do not provide any guarantee on trapdoor privacy, namely the privacy of the messages in trapdoors.

- In the provably secure schemes from [5, 6], a trapdoor contains the target message in clear so that the server knows what the client is search for. For other provably secure schemes in these models, if the target messages are explicitly included in the trapdoors, the proofs will still hold.
- Due to the interdependency, these models fail to capture the unnecessary information leakage from ciphertexts. For example, in the schemes from [5, 6], if a match is found then the server immediately learns the plaintext, otherwise the server learns that the plaintext is different from the target message.

Some security models (e.g. [7]) consider one-wayness property of trapdoors for asymmetric SE schemes. However, one-wayness is a rather weak property and it does not prevent the attacker from learning the client's search patters. For instance, in the provably secure scheme from [7], if the client has submitted two search queries which

[1] The security model from [4] generally assumes that a trapdoor encodes a predicate and a search is in the form of evaluating the encoded predicate on the encrypted plaintext. As a special case, we can assume that the predicate is an equality test for a particular message.

do not result in any match then the attacker knows whether the two search queries are for the same target message or not.

It remains as an open problem to investigate a security model for simultaneously capturing information leakages from both ciphertexts and trapdoors.

1.2 Our Contribution

The contribution of this paper is multi-fold. Firstly, we present two trapdoor privacy definitions (i.e. 2-TRAP-PRIV and poly-TRAP-PRIV) to model the information leakages from the trapdoors in ASE schemes. Due to the nature of ASE schemes (i.e. any entity can encrypt messages by itself), separate modeling of information leakages from ciphertexts and trapdoors is actually equivalent to a hybrid model of the style [4]. Secondly, to facilitate generic constructions using the IBE→ASE framework [3, 8], we introduce two key anonymity properties (i.e. 2-KEY-ANO and poly-KEY-ANO) for IBE that directly lead to the resulting ASE's 2-TRAP-PRIV and poly-TRAP-PRIV properties respectively at the end of a transformation. Both properties may be of independent interest for IBE in other applications. Thirdly, we present a simplified Boyen-Waters scheme and prove that it achieves IBE-IND-CPA, IBE-ANO, and 2-KEY-ANO security in the random oracle model. As a result, we obtain an ASE scheme which achieves computational consistency, ASE-IND-CPA, and 2-TRAP-PRIV security in the random oracle model. Unfortunately, the simplified Boyen-Waters scheme does not achieve poly-KEY-ANO security so that the resulting ASE scheme does not achieve poly-TRAP-PRIV security. Finally, we extend the simplified Boyen-Waters scheme to be based on pairings over composite-order groups and prove that the extended scheme achieves poly-KEY-ANO security without random oracles. This extension may indicate a general approach for other IBE schemes (e.g. the new BB2 scheme from [9]) to achieve the poly-KEY-ANO security.

1.3 Structure of the Paper

The rest of this paper is organized as follows. In Section 2, we first review bilinear groups and some hardness assumptions, then introduce a new assumption, and finally review IBE, ASE, and the transformation from IBE to ASE. In Section 3, we present the new trapdoor privacy definitions for ASE and the corresponding key-anonymity property definitions for IBE. In Section 4, we introduce the simplified Boyen-Waters scheme and prove its security properties. In Section 5, we extend the simplified Boyen-Waters scheme to be based on pairing over composite-order groups and prove its poly-KEY-ANO security. In Section 6, we conclude the paper.

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 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 $\mathcal{A}$ is a probabilistic algorithm we write $\mathsf{a} \leftarrow_\$ \mathcal{A}(\mathsf{i}_1, \mathsf{i}_2, \ldots, \mathsf{i}_n)$ for the action of running $\mathcal{A}$ on inputs $\mathsf{i}_1, \mathsf{i}_2, \ldots, \mathsf{i}_n$ with random coins of its choice, and assigning the result to a. If a is a variable, $|\mathsf{a}|$ denotes the length in bits of its representation.

GAMES. In this paper we use the code-based game-playing language [10]. 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 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 review pairings over prime-order groups [11] and the associated DBDH (Decision Bilinear Diffie-Hellman) and DLIN (Decision linear) assumptions [12], and then review pairings over composite-order groups [13] and introduce the CDDH (Composite Decision Diffie-Hellman) assumption which is weaker than the C3DH (Cpmposite 3-party Diffie-Hellman) assumption made in [14].

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 prime-order p with efficiently-computable group laws.*
- *g is a generator of $\mathbb{G}$.*
- *$\mathbf{e}$ is an efficiently-computable bilinear pairing $\mathbf{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_{\mathsf{T}}$, i.e., a map satisfying the following properties:*
  - *Bilinearity: $\forall \mathsf{a}, \mathsf{b} \in \mathbb{Z}_{\mathsf{p}}, \mathbf{e}(\mathsf{g}^{\mathsf{a}}, \mathsf{g}^{\mathsf{b}}) = \mathbf{e}(\mathsf{g}, \mathsf{g})^{\mathsf{ab}}$;*
  - *Non-degeneracy: $\mathbf{e}(\mathsf{g}, \mathsf{g}) \neq 1$.*

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.*

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

where game $\mathsf{DBDH}_{\Gamma,\mathcal{A}}$ is described in Fig. 1.

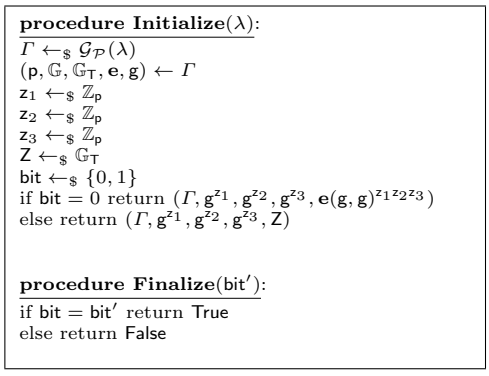

Fig. 1. DBDH Game

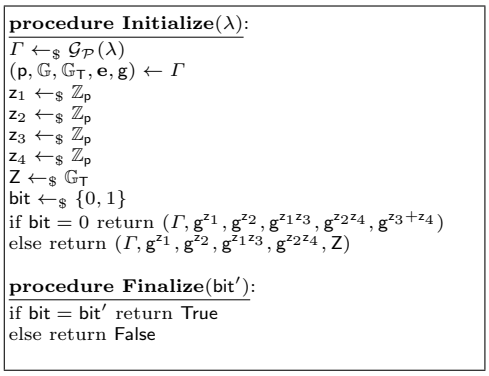

Fig. 2. DLIN Game

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.*

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

where game $\mathsf{DLIN}_{\Gamma,\mathcal{A}}$ is described in Fig. 2.

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 primes, with efficiently computable group laws.*
- g *is a generator of* $\mathbb{G}$*.*
- $\mathbf{e}$ *is an efficiently-computable bilinear pairing* $\mathbf{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_{\mathsf{T}}$*, i.e., a map satisfying the following properties:*
  - *Bilinearity:* $\forall \mathsf{a}, \mathsf{b} \in \mathbb{Z}_{\mathsf{n}}, \mathbf{e}(\mathsf{g}^{\mathsf{a}}, \mathsf{g}^{\mathsf{b}}) = \mathbf{e}(\mathsf{g}, \mathsf{g})^{\mathsf{ab}}$*;*
  - *Non-degeneracy:* $\mathbf{e}(\mathsf{g}, \mathsf{g}) \neq 1$*.*

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, later used in our proofs:

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

Definition 5. *Let $\Gamma = (\mathsf{p}, \mathbb{G}, \mathbb{G}_{\mathsf{T}}, \mathbf{e}, \mathsf{g})$ be the description output by $\mathcal{G}_{\mathcal{C}}(\lambda)$. 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.*

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

where game $\mathsf{CDDH}_{\Gamma,\mathcal{A}}$ is described in Fig. 3.

procedure Initialize(λ):
$(\mathsf{p}, \mathsf{q}, \mathbb{G}, \mathbb{G}_{\mathsf{T}}, \mathbf{e}, \mathsf{g}) \leftarrow_{\$} \mathcal{G}_{\mathcal{C}}(\lambda)$; $\mathsf{n} = \mathsf{pq}$; $\mathsf{g}_{\mathsf{p}} = \mathsf{g}^{\mathsf{q}}$; $\mathsf{g}_{\mathsf{q}} = \mathsf{g}^{\mathsf{p}}$; $\Gamma = (\mathsf{n}, \mathbb{G}, \mathbb{G}_{\mathsf{T}}, \mathbf{e}, \mathsf{g}, \mathsf{g}_{\mathsf{p}}, \mathsf{g}_{\mathsf{q}})$
$\mathsf{x}_1 \leftarrow_{\$} \mathbb{Z}_{\mathsf{n}}$; $\mathsf{X}_1 \leftarrow (\mathsf{g}_{\mathsf{q}})^{\mathsf{x}_1}$; $\mathsf{x}_2 \leftarrow_{\$} \mathbb{Z}_{\mathsf{n}}$; $\mathsf{X}_2 \leftarrow (\mathsf{g}_{\mathsf{q}})^{\mathsf{x}_2}$; $\mathsf{x}_3 \leftarrow_{\$} \mathbb{Z}_{\mathsf{n}}$; $\mathsf{X}_3 \leftarrow (\mathsf{g}_{\mathsf{q}})^{\mathsf{x}_3}$
$\mathsf{a} \leftarrow_{\$} \mathbb{Z}_{\mathsf{n}}$; $\mathsf{R} \leftarrow_{\$} \mathbb{G}$; $\mathsf{b} \leftarrow_{\$} \mathbb{Z}_{\mathsf{n}}$
$\mathsf{bit} \leftarrow_{\$} \{0, 1\}$
if $\mathsf{bit} = 0$ return $(\Gamma, \mathsf{X}_1 \cdot (\mathsf{g}_{\mathsf{p}})^{\mathsf{a}}, \mathsf{X}_2 \cdot (\mathsf{g}_{\mathsf{p}})^{\mathsf{b}}, \mathsf{X}_3 \cdot (\mathsf{g}_{\mathsf{p}})^{\mathsf{ab}})$
else return $(\Gamma, \mathsf{X}_1 \cdot (\mathsf{g}_{\mathsf{p}})^{\mathsf{a}}, \mathsf{X}_2 \cdot (\mathsf{g}_{\mathsf{p}})^{\mathsf{b}}, \mathsf{R})$

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

Fig. 3. CDDH Game

In the CDDH game, the attacker's input is a proper subset of the attacker's input in the C3DH game given in [14], therefore, the CDDH assumption is weaker than the C3DH assumption.

2.2 IBE, ASE, and the Transformation

IBE. 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 public-key space $\mathcal{I}$.

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

The correctness, IBE-IND-CPA, and IBE-ANO (anonymity) properties are formally presented in Appendix I. Informally, the IBE-IND-CPA property guarantees that an attacker cannot distinguish the ciphertexts of $\{\mathsf{m}_0, \mathsf{m}_1\}$ under identity id^* given the access to any $\mathsf{sk}_{\mathsf{id}}$ except for $\mathsf{sk}_{\mathsf{id}^*}$. The IBE-ANO property guarantees that an attacker cannot distinguish the ciphertexts of m under identities $\{\mathsf{id}_0, \mathsf{id}_1\}$ given the access to any $\mathsf{sk}_{\mathsf{id}}$ except those for $\{\mathsf{id}_0, \mathsf{id}_1\}$.

ASE. An ASE scheme $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{PEKS}, \mathsf{Trapdoor}, \mathsf{Test})$ is specified by four polynomial-time algorithms associated with a message space $\mathcal{M}'$.

- $\mathsf{KeyGen}(\lambda)$: On input the security parameter λ, this algorithm returns a private/public key pair $(\mathsf{sk}, \mathsf{pk})$.
- $\mathsf{PEKS}(\mathsf{pk}, \mathsf{w})$: On input a public key pk and a message $\mathsf{w} \in \mathcal{M}'$, this algorithm produces a searchable ciphertext c.
- $\mathsf{Trapdoor}(\mathsf{sk}, \mathsf{w})$: On input a secret key sk and a message $\mathsf{w} \in \mathcal{M}'$, this algorithm outputs a trapdoor tp for w.
- $\mathsf{Test}(\mathsf{pk}, \mathsf{c}, \mathsf{tp})$: On input a public key pk, a searchable ciphertext c and a trapdoor tp, this algorithm outputs either True or False.

The computational consistency and ASE-IND-CPA properties are formally presented in Appendix II. Informally, the ASE-IND-CPA property guarantees that an attacker cannot distinguish the ciphertexts of $\{\mathsf{w}_0, \mathsf{w}_1\}$ given the access to any trapdoor tp_{w} except for $\{\mathsf{tp}_{\mathsf{w}_0}, \mathsf{tp}_{\mathsf{w}_1}\}$.

Remark 1. In this ASE definition, we do not explicitly state the message recovery functionality, namely whether w can be recovered from $\mathsf{Trapdoor}(\mathsf{sk}, \mathsf{w})$ given sk. If message recovery is needed and cannot be done from $\mathsf{Trapdoor}(\mathsf{sk}, \mathsf{w})$ (e.g. in the construction from [3]), then we can easily extend the scheme by (1) asking $\mathsf{KeyGen}(\lambda)$ to output an additional private/public key pair $(\mathsf{sk}', \mathsf{pk}')$ for a standard public key encryption scheme and (2) adding the ciphertext of w under pk' to the output of $\mathsf{PEKS}(\mathsf{pk}, \mathsf{w})$. Such extension does not affect the security properties of the original ASE scheme, so that we do not take message recovery into account in the rest of this paper.

Transformation. Given an IBE scheme $\Pi = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$, there is a generic transformation to obtain an ASE scheme $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{PEKS}, \mathsf{Trapdoor}, \mathsf{Test})$ [3, 8]. The transformation from [8] is rephrased as follows. Note that the message space $\mathcal{M}'$ of the derived ASE scheme is the public-key space $\mathcal{I}$ of the original IBE scheme.

- $\mathsf{KeyGen}(\lambda)$: set $(\mathsf{sk}, \mathsf{pk}) = \mathsf{Setup}(\lambda)$.
- $\mathsf{PEKS}(\mathsf{pk}, \mathsf{w})$: set $\mathsf{c} = (\mathsf{m}, \mathsf{Enc}(\mathsf{pk}, \mathsf{m}, \mathsf{w}))$, where $\mathsf{m} \leftarrow_{\$} \mathcal{M}$.
- $\mathsf{Trapdoor}(\mathsf{sk}, \mathsf{w})$: set $\mathsf{tp} = \mathsf{Extract}(\mathsf{params}, \mathsf{sk}, \mathsf{w})$.
- $\mathsf{Test}(\mathsf{pk}, \mathsf{c}, \mathsf{tp})$: if $\mathsf{m} = \mathsf{Dec}(\mathsf{pk}, \mathsf{c}, \mathsf{tp})$ output True, otherwise output False.

In [8], it is proven that the IBE-IND-CPA and IBE-ANO properties of the IBE scheme lead to the computational consistency and ASE-IND-CPA properties for the derived ASE scheme, respectively.

3 Trapdoor Privacy for ASE

In this section, we define the trapdoor privacy properties for ASE, and then map these properties to new key-anonymity properties for IBE.

3.1 Privacy for ASE

Referring to the *Full Security* model for symmetric SE schemes [4], in order to capture the information leakages from both ciphertexts and trapdoors, we may attempt to adopt a similar approach by asking an attacker to distinguish two ciphertext/trapdoor sequences. However, asymmetric SE differs from symmetric SE in two aspects. One is that an attacker can generate a ciphertext for any message, so that an encryption oracle is unnecessary. The other one is that an attacker can trivially distinguish two trapdoors generated based messages of its choice, so that the challenger should choose the messages for generating the trapdoors in the challenge. Due to these differences, it is not easy to have a single hybrid security model for asymmetric SE, instead we can consider the information leakages in three different scenarios.

1. In the first scenario, we need to consider the privacy for the honestly-generated ciphertexts[2], which do not match any trapdoors received by the attacker. The ASE-IND-CPA property under Definition 13 in the Appendix II is the precise model for this.
2. In the second scenario, we need to consider the privacy for honestly-generated ciphertexts and trapdoors which match with each other. This property is indeed very straightforward to be achieved in the random oracle model (e.g. instead of the original messages, use their hash values as the input to PEKS and $\mathsf{Trapdoor}$). We leave a more comprehensive discussion of this property as a future work.
3. In the third scenario, we need to consider the privacy for the trapdoors, which do not match any honestly-generated ciphertexts. This will be addressed in the rest of this paper.

It is worth emphasizing that, in the security models, the attacker refers to the semi-trusted server in ASE schemes.

3.2 Trapdoor Privacy Definitions for ASE

We first present a trapdoor privacy definition for two trapdoors, as shown in Definition 6. If an ASE scheme is secure under this definition then it guarantees that, given two trapdoors whose plaintexts are randomly chosen by the client, a server cannot determine whether the two trapdoors are generated for the same message or not.

Definition 6. *An ASE scheme $\mathcal{E}$ has 2-trapdoor privacy (*$2\text{-}\mathsf{TRAP}\text{-}\mathsf{PRIV}$*) if, for every legitimate PPT adversary $\mathcal{A}$, the following definition of advantage is negligible in λ*

$$\mathbf{Adv}^{2\text{-}\mathsf{TRAP}\text{-}\mathsf{PRIV}}_{\mathcal{E},\mathcal{A}}(\lambda) := 2 \cdot \Pr[2\text{-}\mathsf{TRAP}\text{-}\mathsf{PRIV}_{\mathcal{E},\mathcal{A}}(\lambda) \Rightarrow \mathsf{True}] - 1,$$

where game $2\text{-}\mathsf{TRAP}\text{-}\mathsf{PRIV}_{\mathcal{E},\mathcal{A}}$ *is described in Fig. 4.*

[2] Honestly-generated ciphertexts refer to those generated by honest senders. They are in contrast to the ciphertexts generated by the attacker itself.

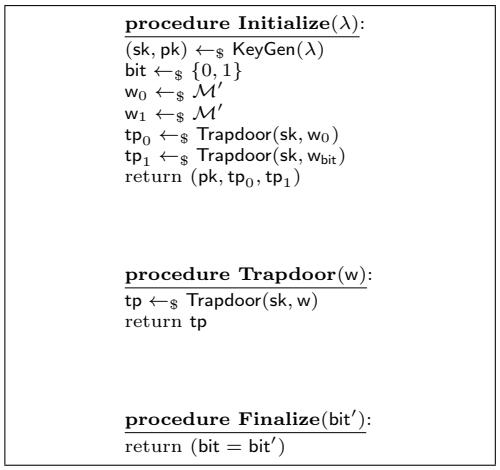

Fig. 4. 2-TRAP-PRIV$_{\mathcal{E},\mathcal{A}}$ Game

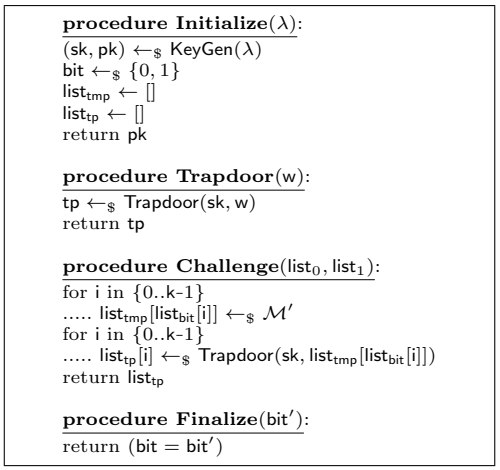

Fig. 5. poly-TRAP-PRIV$_{\mathcal{E},\mathcal{A}}$ Game

Under this definition, a natural question is "with a secure ASE scheme, what the server will learn from more than two trapdoors which do not match with any honestly-generated ciphertexts?". Unfortunately, it does not provide any guarantee in this situation, due to the following fact: generating new trapdoors for the attacker requires the knowledge of both the private key and the underlying messages, and this knowledge may be leaked to the attacker (through the new trapdoors) unless these new trapdoors can be generated by re-randomizing existing ones. This is illustrated by our construction in Section 4.2. Therefore, it makes sense to consider the privacy for any polynomial number of trapdoors, as done in Definition 7 below.

Definition 7. *An ASE scheme $\mathcal{E}$ has poly-trapdoor privacy (*poly-TRAP-PRIV*) if, for every legitimate PPT adversary $\mathcal{A}$ and any L ($L = |\mathsf{list}_0| = |\mathsf{list}_1|$ is any polynomial in λ), the following definition of advantage is negligible in λ*

$$\mathbf{Adv}^{\text{poly-TRAP-PRIV}}_{\mathcal{E},\mathcal{A}}(\lambda) := 2 \cdot \Pr[\text{poly-TRAP-PRIV}_{\mathcal{E},\mathcal{A}}(\lambda) \Rightarrow \mathsf{True}] - 1,$$

where game poly-TRAP-PRIV$_{\mathcal{E},\mathcal{A}}$ *is described in Fig. 5. The adversary is legitimate if it only calls Challenge once in the game.*

Remark 2. It is easy to see that both trapdoor privacy definitions are only achievable when the message space $\mathcal{M}'$ is not polynomial size. Otherwise, an attacker can trivially generate ciphertexts for all messages and then figure out the messages in the trapdoors by running the Test algorithm.

3.3 Key Anonymity Definitions for IBE

Referring to the generic transformation from IBE to ASE as described in Section 2.2, the two trapdoor privacy properties motivate two new key anonymity properties properties for IBE. It is clear that these key anonymity properties lead to the trapdoor privacy properties defined in Section 3.2.

Definition 8. *An IBE scheme Π has 2-key anonymity (*2-KEY-ANO*) if, for every legitimate PPT adversary $\mathcal{A}$, the following definition of advantage is negligible in λ*

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

where game 2-KEY-ANO$_{\Pi,\mathcal{A}}$ *is described in Fig. 6.*

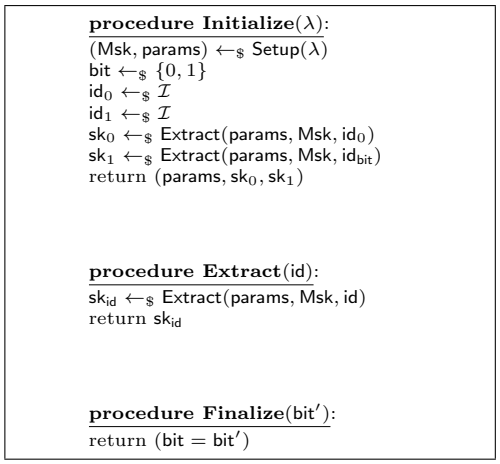

Fig. 6. 2-KEY-ANO$_{\Pi,\mathcal{A}}$ Game

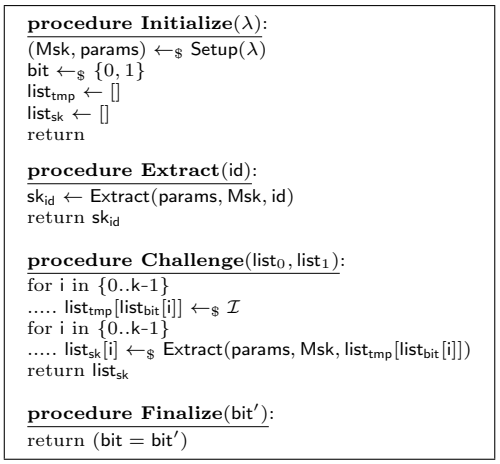

Fig. 7. poly-KEY-ANO$_{\Pi,\mathcal{A}}$ Game

Definition 9. *An IBE scheme Π is has poly-key anonymity (*poly-KEY-ANO*) if, for every legitimate PPT adversary $\mathcal{A}$, the following definition of advantage is negligible in λ*

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

where game poly-KEY-ANO$_{\Pi,\mathcal{A}}$ *is described in Fig. 7. The adversary is legitimate if it only calls Challenge once in the game.*

Remark 3. Similar to the trapdoor privacy properties for ASE, both key anonymity definitions are only achievable when the public-key space $\mathcal{I}$ is not polynomial size.

Besides achieving the trapdoor privacy properties for ASE in the generic transformation, the key-anonymity properties for IBE may be of independent interest. For instances, when IBE is used for enforcing access control policies, even if some entities' keys are compromised these entities' identities can still be hidden from the attacker.

3.4 Remarks on Trapdoor Privacy Definitions

In the same direction as in the *Full Security* definition for symmetric SE schemes [4], we can also have a hybrid model with the security game in Fig. 8 for both the trapdoor privacy and the ASE-IND-CPA privacy. However, based on a similar argument to that in [15] where the security of encrypting one message is proven to be equivalent to encrypting multiple messages, we can show that if an ASE scheme is secure under Definition 7 and Definition 13 then it is also secure under the hybrid definition.

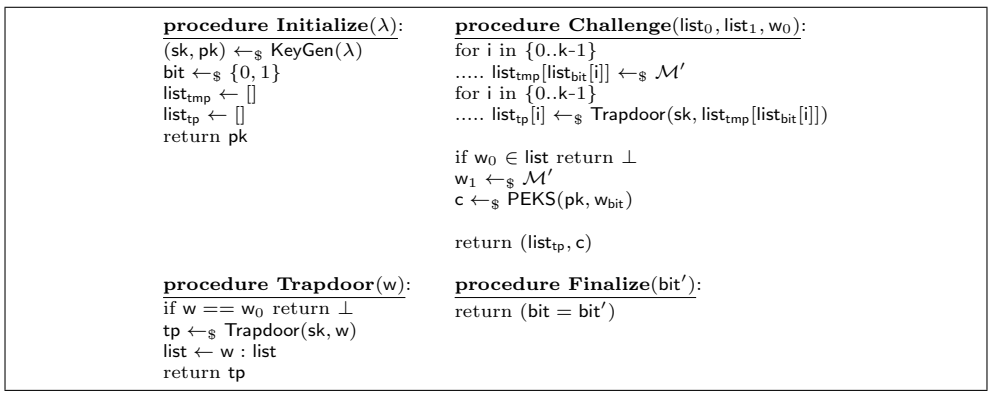

Fig. 8. Hybrid Security Game

In Definition 6 and 7 for ASE, we have assumed that the messages in the challenge are chosen by the challenger uniformly at random. To further enhance the definitions, we may let the attacker specify some relationships that the messages should satisfy. For instance, in Definition 6, an attacker can require that $\mathsf{w}_1 = \mathsf{w}_0+1$. This observation also applies to Definition 8 and 9 for IBE. We leave a more comprehensive discussion about these enhancements as a future work. Nevertheless, in the random oracle model, the enhanced definitions may be equivalent to what we have defined.

4 2-KEY-ANO Secure IBE

In this section, we design a 2-KEY-ANO secure IBE scheme. Based on this IBE scheme, it is trivial to obtain a 2-TRAP-PRIV secure ASE through the generic transformation described in Section 2.2, hence, we skip the details here.

4.1 Concise Survey of Anonymous IBE

As mentioned in Section 2.2, the IBE-IND-CPA and IBE-ANO properties are essential to achieve the computational consistency and ASE-IND-CPA properties in the generic transformation. Therefore, we aim at IBE schemes which achieve IBE-IND-CPA, IBE-ANO, 2-TRAP-PRIV, and poly-KEY-ANO properties.

	IBE-IND-CPA	IBE-ANO	2-TRAP-PRIV	poly-TRAP-PRIV
[11]	Yes (BDH & RO)	Yes(BDH & RO)	No	No
[16]	Yes (DBDH)	No	Maybe	Maybe
BB1 [17]	Yes (DBDH & sID)	No	Maybe	No
BB2 [17]	Yes (DBDHI & sID)	No	Maybe	Maybe
[18]	Yes (DBDH)	No	Maybe	No
[19]	Yes (ABDHE)	Yes (ABDHE)	No	No
[20]	Yes (DBDH & sID)	Yes (DLIN & sID)	Maybe	Maybe
[21]	Yes (LBDH) & RO	Maybe	No	No
[14]	Yes (C3DH & sID)	Yes (C3DH & sID)	Maybe	Maybe
New BB1 [9]	Yes (BDDHI)	No	Maybe	Maybe
New BB2 [9]	Yes (BDDHI)	Yes (BDDHI)	Maybe	No
[22]	Yes (Decision $\mathcal{P}$-BDH & sID)	Yes (Decision $\mathcal{P}$-BDH & sID)	Maybe	Maybe

Table 1. IBE Survey

In Table 1, we survey the existing IBE schemes with respect to their security properties. The "No" means that the scheme does not achieve the property, and the "Maybe" indicates that we have neither an attack nor a proof. From the survey results, the schemes from [14, 20, 22] are promising, if we can prove the properties labeled with "Maybe". We based our solutions on the scheme from [20] for two reasons: its security is based on relatively more standard assumptions, and it only assumes symmetric pairing over prime-order groups.

4.2 2-KEY-ANO Secure IBE

To eliminate the selective-ID constraint of the Boyen-Waters scheme [20], we adopt the random oracle approach introduced in the full paper of [17], by replacing the identities with their hash values in the IBE algorithms. Furthermore, we simplify the resulted scheme by removing two elements from the public parameters and all private keys, and obtain the final scheme in Fig. 9. Compared with the original scheme, the simplified Boyen-Waters scheme saves two exponentiations in the Extract and Enc algorithms, and saves two pairing computations in the Dec algorithm.

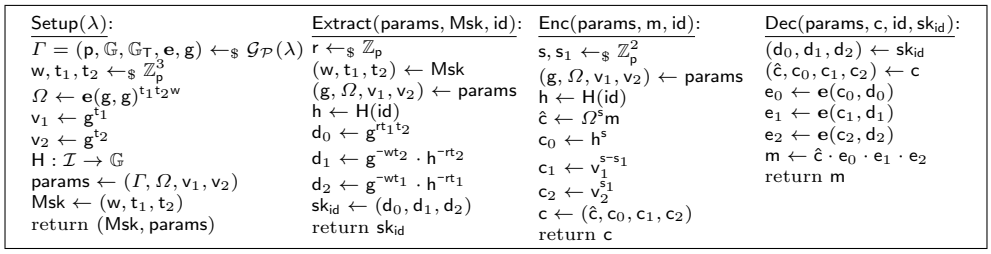

Fig. 9. Simplified Boyen-Waters Scheme

Due to the simplification, we provide new proofs for the IBE-IND-CPA, IBE-ANO, 2-TRAP-PRIV security properties in the random oracle model.

Theorem 1. *The simplified Boyen-Waters scheme scheme Π [Fig. 9] is IBE-IND-CPA secure under Definition 10 in the random oracle model assuming DBDH is intractable.*

Proof. Game_0 is game $\mathsf{IBE\text{-}IND\text{-}CPA}_{\Pi,\mathcal{A}}$ [Fig. 17], where $\mathcal{A}$ is any legitimate PPT adversary. In Game_1, we set $\hat{c} \leftarrow_\$ \mathbb{G}_T$ instead of computing its value. We now show that the existence of $\mathcal{A}$ that can successfully distinguish between Game_0 and Game_1 contradicts the DBDH assumption. More precisely, we construct a simulator $\mathcal{S}_0$ [Figure 10] that interpolates between Game_0 and Game_1 by playing game DBDH [Fig. 1]. The hash function H is modelled as a random oracle and we assume, without loss of generality, that $\mathcal{A}$ always asks for the hash value of id before querying id to oracles Extract and Real-or-Random. Furthermore, for simplicity of exposition, we assume that $\mathcal{A}$ places *exactly* q queries, where q is bounded by some polynomial in the security parameter λ, since $\mathcal{A}$ is required to run in polynomial-time. Simulator $\mathcal{S}_0$ randomly tries to guess which query $i \in \{0, q\text{-}1\}$ contains the id on which adversary $\mathcal{A}$ asks to be challenged. When i is *not* sucessfully guessed, $\mathcal{S}_0$ simply aborts. But when it is, which happens with probability $\frac{1}{q}$, $\mathcal{S}_0$ perfectly simulates Game_0 if Z is of the form $e(g,g)^{z_1 z_2 z_3}$, and perfectly simulates Game_1 if Z is just a random element of $\mathbb{G}_T$. Notice that this implies $w = z_1 z_2$ and $s = z_3$.
Let $id^\star$ be the id chosen by $\mathcal{A}$ for the challenge. Random function H is consistently computed: if queried twice on the same id, the same result is returned. When $\mathcal{S}_0$ successfully completes its simulation, the function is set to $(g^{z_1})^x$ for every id but $id^\star$, and to g^x in this particular case, where x is a random value sampled from $\mathbb{Z}_p$.
Challenge is well formed, as well as secret keys for random exponents $r' = \frac{r - z_2}{x}$, where x here is the value used to compute the hash of the corresponding id and r is sampled from $\mathbb{Z}_p$. For completeness, we present the equalities between the original expressions and those computed by the simulator[3].

$$d_0 = g^{r' t_1 t_2} = g^{\frac{r - z_2}{x} t_1 t_2} = g^{\frac{r t_1 t_2}{x}} \cdot g^{\frac{-z_2 t_1 t_2}{x}} = g^{\frac{r t_1 t_2}{x}} \cdot Z_2^{\frac{-t_1 t_2}{x}}$$
$$d_1 = g^{-w t_2} \cdot h^{-r' t_2} = g^{-z_1 z_2 t_2} \cdot \left[(g^{z_1})^x\right]^{-\frac{r - z_2}{x} t_2} = Z_1^{-r t_2}$$
$$d_2 = g^{-w t_1} \cdot h^{-r' t_1} = g^{-z_1 z_2 t_1} \cdot \left[(g^{z_1})^x\right]^{-\frac{r - z_2}{x} t_1} = Z_1^{-r t_1}$$

$$\hat{c} = \Omega^s \cdot m = \left[e(g,g)^{t_1 t_2 w}\right]^s \cdot m = \left[e(g,g)^{t_1 t_2 z_1 z_2}\right]^{z_3} \cdot m = Z^{t_1 t_2} \cdot m$$
$$c_0 = h^s = (g^x)^{z_3} = Z_3^x$$
$$c_1 = v_1^{s - s_1} = (g_1^t)^{z_3 - s_1} = (Z_3 \cdot g^{-s_1})^{t_1}$$
$$c_2 = v_2^{s_1}$$

In Game_1, the challenge ciphertext does not depend on m_{bit}. Therefore, we have that $\mathbf{Adv}^{\mathsf{IBE\text{-}IND\text{-}CPA}}_{\Pi,\mathcal{A}}(\lambda) = \frac{1}{q} \cdot \mathbf{Adv}_{\mathsf{DBDH},\mathcal{S}_0}$, which concludes our proof. □

[3] For $\hat{c}$ we used the case where $Z = e(g,g)^{z_1 z_2 z_3}$, which corresponds to the simulation of Game_0.

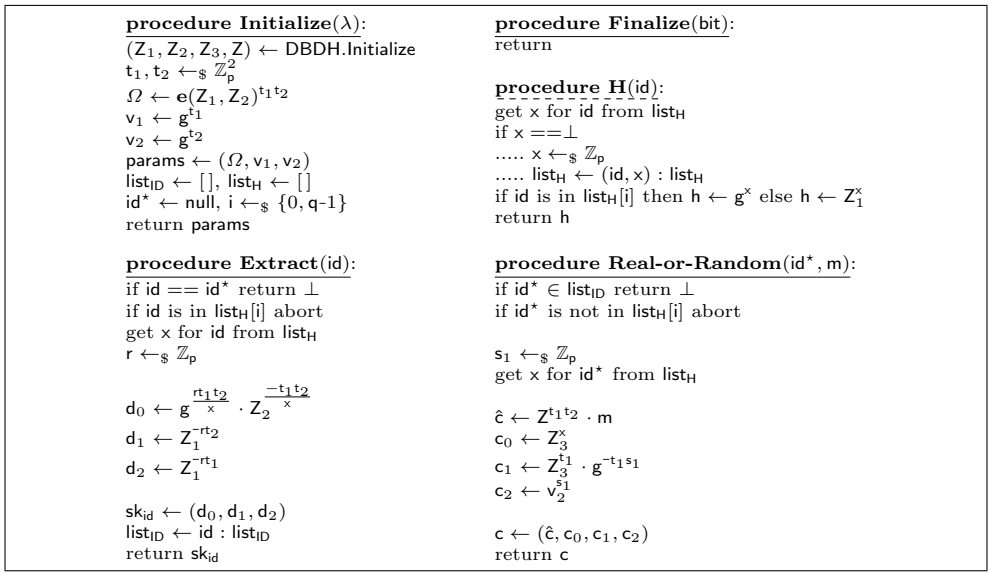

Fig. 10. Simulator $\mathcal{S}_0$ interpolates between Game_0 and Game_1 by playing game DBDH.

Theorem 2. *The simplified Boyen-Waters scheme Π [Fig. 9] is IBE-ANO secure under Definition 11 in the random oracle model assuming DBDH and DLIN are intractable.*

Proof. By applying the reduction of Theorem 1, we start with Game_1, which sets $\hat{c} \leftarrow_\$ \mathbb{G}_\mathsf{T}$. In Game_2, instead of computing its value, we set $\mathsf{c}_1 \leftarrow_\$ \mathbb{G}$. For $\mathcal{A}$ to successfully distinguish between Game_1 and Game_2, the DLIN assumption would have to be tractable. Formally, we show this by constructing a simulator $\mathcal{S}_1$ [Fig. 11] that interpolates between Game_1 and Game_2 by playing game DLIN [Fig. 2]. As in Theorem 1, the hash function H is modelled as a random oracle. Without loss of generality, we assume that $\mathcal{A}$ places *exactly* q queries and always asks for the hash value of id before querying id to Extract and Real-or-Random. Employing the same strategy as before, simulator $\mathcal{S}_1$ randomly tries to guess which query $\mathsf{i} \in \{0, \mathsf{q}\text{-}1\}$ contains the id on which $\mathcal{A}$ asks to be challenged. When i is sucessfully guessed, which happens with probability $\frac{1}{\mathsf{q}}$, $\mathcal{S}_1$ perfectly simulates Game_1 if Z is of the form $\mathsf{g}^{\mathsf{z}_3+\mathsf{z}_4}$, and perfectly simulates Game_2 otherwise. This implies that $\mathsf{t}_1 = \mathsf{z}_1$ and $\mathsf{t}_2 = \mathsf{z}_2$. Random function H is set to $(\mathsf{g}^{\mathsf{z}_2})^\mathsf{x}$ for every id but the id chosen for the challenge, which is set to g^x, where x is a random value sampled from $\mathbb{Z}_\mathsf{p}$. Challenge is well formed, as well as secret keys for random exponents $\mathsf{r}' = \frac{\mathsf{r}}{\mathsf{z}_2}$, where r is sampled from $\mathbb{Z}_\mathsf{p}$. Finally, notice that $\mathsf{s} = \mathsf{z}_3 + \mathsf{z}_4$ and $\mathsf{s}_1 = \mathsf{z}_4$, and that t_1, t_2, r', s and s_1 are uniformly distributed over $\mathbb{Z}_\mathsf{p}$, as they should. For completeness, we present the equalities between the original expressions and those computed by the simulator[4].

$$\mathsf{d}_0 = \mathsf{g}^{\mathsf{r}'\mathsf{t}_1\mathsf{t}_2} = \mathsf{g}^{\frac{\mathsf{r}}{\mathsf{z}_2}\mathsf{z}_1\mathsf{z}_2} = (\mathsf{g}^{\mathsf{z}_1})^\mathsf{r} = \mathsf{Z}_1^\mathsf{r}$$
$$\mathsf{d}_1 = \mathsf{g}^{-\mathsf{w}\mathsf{t}_2} \cdot \mathsf{h}^{-\mathsf{r}'\mathsf{t}_2} = \mathsf{g}^{-\mathsf{w}\mathsf{z}_2} \cdot \left[(\mathsf{g}^{\mathsf{z}_2})^\mathsf{x}\right]^{-\frac{\mathsf{r}}{\mathsf{z}_2}\mathsf{z}_2} = \mathsf{Z}_2^{-\mathsf{w}} \cdot \mathsf{Z}_2^{-\mathsf{x}\mathsf{r}} = \mathsf{Z}_2^{-\mathsf{w}-\mathsf{x}\mathsf{r}}$$
$$\mathsf{d}_2 = \mathsf{g}^{-\mathsf{w}\mathsf{t}_1} \cdot \mathsf{h}^{-\mathsf{r}'\mathsf{t}_1} = \mathsf{g}^{-\mathsf{w}\mathsf{z}_1} \cdot \left[(\mathsf{g}^{\mathsf{z}_2})^\mathsf{x}\right]^{-\frac{\mathsf{r}}{\mathsf{z}_2}\mathsf{z}_1} = \mathsf{Z}_1^{-\mathsf{w}} \cdot \mathsf{Z}_1^{-\mathsf{x}\mathsf{r}} = \mathsf{Z}_1^{-\mathsf{w}-\mathsf{x}\mathsf{r}}$$

$$\mathsf{c}_0 = \mathsf{h}^\mathsf{s} = (\mathsf{g}^\mathsf{x})^{\mathsf{z}_3+\mathsf{z}_4} = \mathsf{Z}^\mathsf{x}$$
$$\mathsf{c}_1 = \mathsf{v}_1^{\mathsf{s}-\mathsf{s}_1} = \left(\mathsf{g}^{\mathsf{z}_1}\right)^{(\mathsf{z}_3+\mathsf{z}_4)-\mathsf{z}_4} = \mathsf{Z}_{13}$$
$$\mathsf{c}_2 = \mathsf{v}_2^{\mathsf{s}_1} = (\mathsf{g}^{\mathsf{z}_2})^{\mathsf{z}_4} = \mathsf{Z}_{24}$$

In Game_2, the challenge ciphertext does not depend on the receiver's identity. Therefore, we have that $\mathbf{Adv}^{\mathsf{IBE\text{-}ANO}}_{\Pi,\mathcal{A}}(\lambda) = \frac{1}{\mathsf{q}} \cdot \mathbf{Adv}_{\mathsf{DBDH},\mathcal{S}_0} + \frac{1}{\mathsf{q}} \cdot \mathbf{Adv}_{\mathsf{DLIN},\mathcal{S}_1}$, which concludes our proof. □

[4] For c_0 we used the case where $\mathsf{Z} = \mathsf{g}^{\mathsf{z}_3+\mathsf{z}_4}$, which corresponds to the simulation of Game_1.

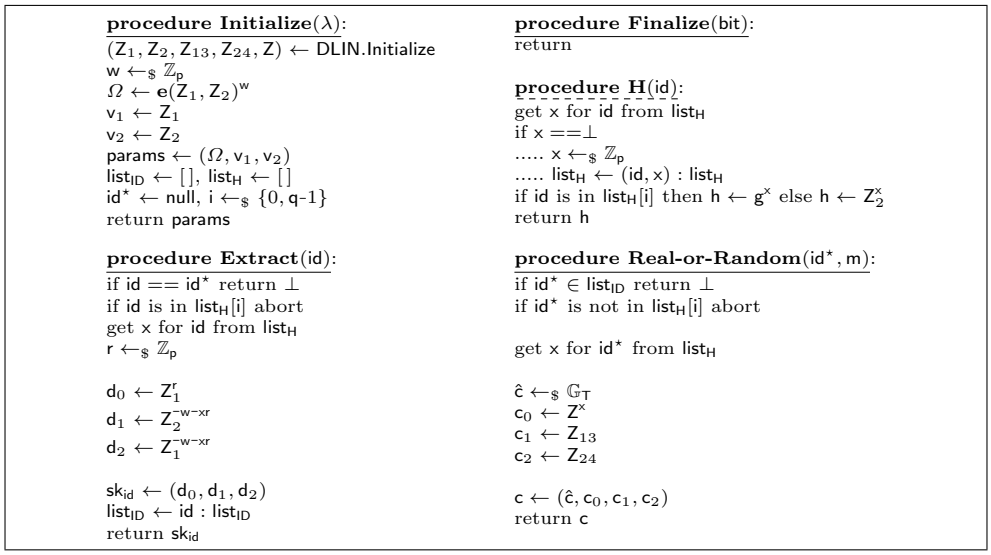

Fig. 11. Simulator $\mathcal{S}_1$ interpolates between Game_1 and Game_2 by playing game DLIN.

Theorem 3. *The simplified Boyen-Waters scheme Π [Fig. 9] is* 2-KEY-ANO *secure under Definition 8 in the random oracle model assuming DLIN is intractable.*

Proof. Let $\mathcal{A}$ be any legitimate PPT adversary in game 2-KEY-ANO$_{\Pi,\mathcal{A}}$ [Fig. 6], where Π is the scheme we presented in Fig. 9. By building a simulator $\mathcal{S}_2$ [Fig. 12] that plays game DLIN [Fig. 2] and simulates game 2-KEY-ANO$_{\Pi,\mathcal{A}}$ in such a way that $\mathcal{A}$'s guess can be forward to game DLIN, 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 modelled 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 uniformily 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 uniformily distributed over $\mathbb{G}$, and $\mathsf{r}^\star$ and r° are uniformily 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.

$$\mathsf{d}_0^\star = \mathsf{g}^{\mathsf{r}^\star \mathsf{t}_1 \mathsf{t}_2} = \mathsf{g}^{\frac{\mathsf{b}}{(\mathsf{z}_1)^2}\mathsf{z}_1\mathsf{z}_1\mathsf{a}} = \mathsf{g}^{\mathsf{ab}}$$
$$\mathsf{d}_1^\star = \mathsf{g}^{-\mathsf{w}\mathsf{t}_2} \cdot \mathsf{h}^{-\mathsf{r}^\star \mathsf{t}_2} = \mathsf{g}^{-\frac{\mathsf{z}_3\mathsf{b}}{\mathsf{z}_1}\mathsf{z}_1\mathsf{a}} \cdot (\mathsf{g}^{\mathsf{z}_1\mathsf{z}_4})^{-\frac{\mathsf{b}}{(\mathsf{z}_1)^2}\mathsf{z}_1\mathsf{a}} = (\mathsf{g}^{-\mathsf{ab}})^{\mathsf{z}_3} \cdot (\mathsf{g}^{-\mathsf{ab}})^{\mathsf{z}_4} = \mathsf{Z}^{-\mathsf{ab}}$$
$$\mathsf{d}_1^\star = \mathsf{g}^{-\mathsf{w}\mathsf{t}_1} \cdot \mathsf{h}^{-\mathsf{r}^\star \mathsf{t}_1} = \mathsf{g}^{-\frac{\mathsf{z}_3\mathsf{b}}{\mathsf{z}_1}\mathsf{z}_1} \cdot (\mathsf{g}^{\mathsf{z}_1\mathsf{z}_4})^{-\frac{\mathsf{b}}{(\mathsf{z}_1)^2}\mathsf{z}_1} = (\mathsf{g}^{-\mathsf{b}})^{\mathsf{z}_3} \cdot (\mathsf{g}^{-\mathsf{b}})^{\mathsf{z}_4} = \mathsf{Z}^{-\mathsf{b}}$$

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

Therefore, we have that $\mathbf{Adv}^{\text{2-KEY-ANO}}_{\Pi,\mathcal{A}}(\lambda) = \mathbf{Adv}_{\mathsf{DLIN},\mathcal{S}_2}$, which concludes our proof. □

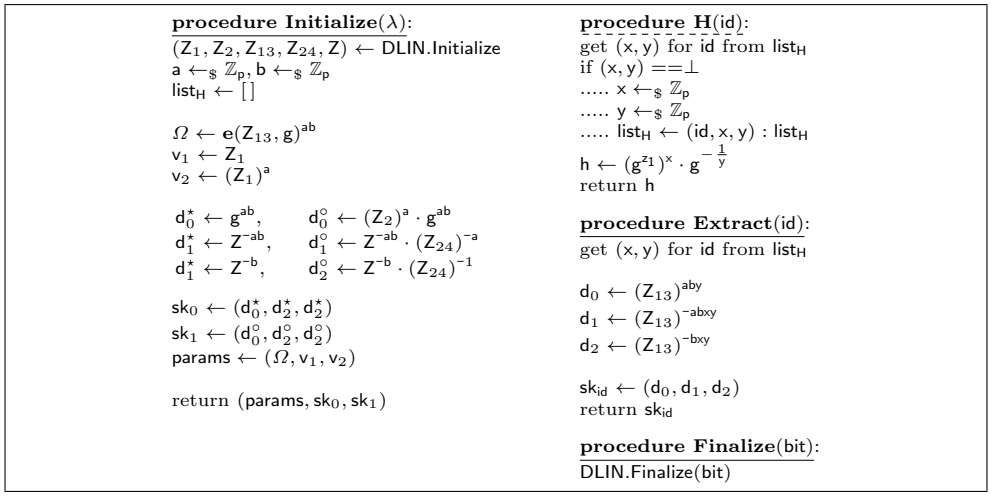

Fig. 12. Simulator $\mathcal{S}_2$ forwards $\mathcal{A}$'s guess from 2-KEY-ANO$_{\Pi,\mathcal{A}}$ to game DLIN.

4.3 Non *poly*-KEY-ANO Security

Standard left-or-right (and 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 [15]. One might be tempted to think that the same hybrid argument also applies to 2-KEY-ANO. However, this argument does *not* because we can show that an adversary can still easily distinguish patterns when more than two keys are issued.

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 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 trapdoors are very likely to have been extracted for the same id[5]. If the result is false, then the tuple is definitely of the form

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

$(\mathsf{Extract}(\mathsf{id}_0), \mathsf{Extract}(\mathsf{id}_0), \mathsf{Extract}(\mathsf{id}_1))$. For completeness, we show this by expanding and simplifying the above expression.

$$\begin{aligned}
\mathbf{e}(\frac{\mathsf{d}_{10}}{\mathsf{d}_{00}}, \frac{\mathsf{d}_{21}}{\mathsf{d}_{01}}) = \mathbf{e}(\frac{\mathsf{d}_{00}}{\mathsf{d}_{20}}, \frac{\mathsf{d}_{01}}{\mathsf{d}_{11}}) &\Leftrightarrow \\
\mathbf{e}(\frac{\mathsf{g}^{r_1 t_1 t_2}}{\mathsf{g}^{r_0 t_1 t_2}}, \frac{\mathsf{g}^{-wt_2} \cdot \mathsf{h}_2^{-r_2 t_2}}{\mathsf{g}^{-wt_2} \cdot \mathsf{h}_0^{-r_0 t_2}}) = \mathbf{e}(\frac{\mathsf{g}^{r_0 t_1 t_2}}{\mathsf{g}^{r_2 t_1 t_2}}, \frac{\mathsf{g}^{-wt_2} \cdot \mathsf{h}_0^{-r_0 t_2}}{\mathsf{g}^{-wt_2} \cdot \mathsf{h}_1^{-r_1 t_2}}) &\Leftrightarrow \\
\mathbf{e}(\frac{\mathsf{g}^{r_1 t_1 t_2}}{\mathsf{g}^{r_0 t_1 t_2}}, \frac{\mathsf{h}_2^{-r_2 t_2}}{\mathsf{h}_0^{-r_0 t_2}}) = \mathbf{e}(\frac{\mathsf{g}^{r_0 t_1 t_2}}{\mathsf{g}^{r_2 t_1 t_2}}, \frac{\mathsf{h}_0^{-r_0 t_2}}{\mathsf{h}_0^{-r_1 t_2}}) &\Leftrightarrow \\
\mathbf{e}(\mathsf{g}^{(r_1 - r_0)}, \mathsf{h}_0^{r_0} \cdot \mathsf{h}_2^{-r_2})^{t_1 (t_2)^2} = \mathbf{e}(\mathsf{g}^{(r_0 - r_2)}, \mathsf{h}_0^{(r_1 - r_0)})^{t_1 (t_2)^2} &\Leftrightarrow \\
\mathbf{e}(\mathsf{g}, \mathsf{h}_0^{r_0} \cdot \mathsf{h}_2^{-r_2}) = \mathbf{e}(\mathsf{g}, \mathsf{h}_0^{(r_0 - r_2)}) &\Leftrightarrow \\
\mathsf{h}_2 = \mathsf{h}_0 &
\end{aligned}$$

It is now clear that the simplified Boyen-Waters scheme fails to achieve the *poly*-KEY-ANO security.

5 *poly*-KEY-ANO Secure IBE

As shown in Fig. 13, we extend the simplified Boyen-Waters IBE scheme described in Fig. 9 to be based on pairings over composite-order groups. The extension is very simple: let all the parameters in the original scheme be from the subgroup $\mathbb{G}_\mathsf{p}$ (generated by g_p) and randomize each element of the extracted secret key by a random element from the subgroup $\mathbb{G}_\mathsf{q}$ (generated by g_q). Note that the message space is $\mathbb{G}_\mathsf{T}$.

$\mathsf{Setup}(1^\lambda)$:	$\mathsf{Extract}(\mathsf{params}, \mathsf{Msk}, \mathsf{id})$:	$\mathsf{Enc}(\mathsf{params}, \mathsf{m}, \mathsf{id})$:	$\mathsf{Dec}(\mathsf{params}, \mathsf{c}, \mathsf{id}, \mathsf{sk}_\mathsf{id})$:
$(\mathsf{p}, \mathsf{q}, \mathbb{G}, \mathbb{G}_\mathsf{T}, \mathbf{e}, \mathsf{g}) \leftarrow_\$ \mathcal{G}_\mathcal{C}(\lambda)$	$(\mathsf{w}, \mathsf{t}_1, \mathsf{t}_2) \leftarrow \mathsf{Msk}$	$(\Gamma, \Omega, \mathsf{v}_1, \mathsf{v}_2) \leftarrow \mathsf{params}$	$(\mathsf{d}_0, \mathsf{d}_1, \mathsf{d}_2) \leftarrow \mathsf{sk}_\mathsf{id}$
$\mathsf{n} \leftarrow \mathsf{pq}$; $\mathsf{g}_\mathsf{p} \leftarrow \mathsf{g}^\mathsf{q}$; $\mathsf{g}_\mathsf{q} \leftarrow \mathsf{g}^\mathsf{p}$	$(\Gamma, \Omega, \mathsf{v}_1, \mathsf{v}_2) \leftarrow \mathsf{params}$	$(\mathsf{n}, \mathbb{G}, \mathbb{G}_\mathsf{T}, \mathbf{e}, \mathsf{g}, \mathsf{g}_\mathsf{p}, \mathsf{g}_\mathsf{q}) \leftarrow \Gamma$	$(\hat{\mathsf{c}}, \mathsf{c}_0, \mathsf{c}_1, \mathsf{c}_2) \leftarrow \mathsf{c}$
$\Gamma \leftarrow (\mathsf{n}, \mathbb{G}, \mathbb{G}_\mathsf{T}, \mathbf{e}, \mathsf{g}, \mathsf{g}_\mathsf{p}, \mathsf{g}_\mathsf{q})$	$(\mathsf{n}, \mathbb{G}, \mathbb{G}_\mathsf{T}, \mathbf{e}, \mathsf{g}, \mathsf{g}_\mathsf{p}, \mathsf{g}_\mathsf{q}) \leftarrow \Gamma$	$\mathsf{s}, \mathsf{s}_1 \leftarrow_\$ \mathbb{Z}_\mathsf{n}$	$\mathsf{e}_0 \leftarrow \mathbf{e}(\mathsf{c}_0, \mathsf{d}_0)$
$\mathsf{w}, \mathsf{t}_1, \mathsf{t}_2 \leftarrow_\$ \mathbb{Z}_\mathsf{n}$	$\mathsf{r} \leftarrow_\$ \mathbb{Z}_\mathsf{n}$	$\mathsf{h} \leftarrow \mathsf{H}(\mathsf{id})$	$\mathsf{e}_1 \leftarrow \mathbf{e}(\mathsf{c}_1, \mathsf{d}_1)$
$\Omega \leftarrow \mathbf{e}(\mathsf{g}_\mathsf{p}, \mathsf{g}_\mathsf{p})^{t_1 t_2 w}$	$\mathsf{x}_0, \mathsf{x}_1, \mathsf{x}_2 \leftarrow_\$ \mathbb{G}_\mathsf{q}$	$\hat{\mathsf{c}} \leftarrow \Omega^\mathsf{s} \mathsf{m}$	$\mathsf{e}_2 \leftarrow \mathbf{e}(\mathsf{c}_2, \mathsf{d}_2)$
$\mathsf{v}_1 \leftarrow \mathsf{g}_\mathsf{p}^{t_1}$	$\mathsf{h} \leftarrow \mathsf{H}(\mathsf{id})$	$\mathsf{c}_0 \leftarrow \mathsf{h}^\mathsf{s}$	$\mathsf{m} \leftarrow \hat{\mathsf{c}} \cdot \mathsf{e}_0 \cdot \mathsf{e}_1 \cdot \mathsf{e}_2$
$\mathsf{v}_2 \leftarrow \mathsf{g}_\mathsf{p}^{t_2}$	$\mathsf{d}_0 \leftarrow \mathsf{x}_0 \cdot \mathsf{g}_\mathsf{p}^{r t_1 t_2}$	$\mathsf{c}_1 \leftarrow \mathsf{v}_1^{\mathsf{s}-\mathsf{s}_1}$	return m
$\mathsf{params} \leftarrow (\Gamma, \Omega, \mathsf{v}_1, \mathsf{v}_2)$	$\mathsf{d}_1 \leftarrow \mathsf{x}_1 \cdot \mathsf{g}_\mathsf{p}^{-wt_2} \cdot \mathsf{h}^{-rt_2}$	$\mathsf{c}_2 \leftarrow \mathsf{v}_2^{\mathsf{s}_1}$	
$\mathsf{Msk} \leftarrow (\mathsf{w}, \mathsf{t}_1, \mathsf{t}_2)$	$\mathsf{d}_2 \leftarrow \mathsf{x}_2 \cdot \mathsf{g}_\mathsf{p}^{-wt_1} \cdot \mathsf{h}^{-rt_1}$	$\mathsf{c} \leftarrow (\hat{\mathsf{c}}, \mathsf{c}_0, \mathsf{c}_1, \mathsf{c}_2)$	
return $(\mathsf{Msk}, \mathsf{params})$	$\mathsf{sk} \leftarrow (\mathsf{d}_0, \mathsf{d}_1, \mathsf{d}_2)$	return c	
	return sk		

Fig. 13. Extended Simplified Boyen-Waters Scheme Π'

In this extended IBE scheme, the decryption algorithm remains correct, since

$$\begin{aligned}
&\mathbf{e}(\mathsf{h}^\mathsf{s}, \mathsf{x}_0 \cdot \mathsf{g}_\mathsf{p}^{r t_1 t_2}) = \mathbf{e}(\mathsf{h}^\mathsf{s}, \mathsf{g}_\mathsf{p}^{r t_1 t_2}) \\
&\mathbf{e}(\mathsf{v}_1^{\mathsf{s}-\mathsf{s}_1}, \mathsf{x}_1 \cdot \mathsf{g}_\mathsf{p}^{-wt_2} \cdot \mathsf{h}^{-rt_2}) = \mathbf{e}(\mathsf{v}_1^{\mathsf{s}-\mathsf{s}_1}, \mathsf{g}_\mathsf{p}^{-wt_2} \cdot \mathsf{h}^{-rt_2}) \\
&\mathbf{e}(\mathsf{v}_2^{\mathsf{s}_1}, \mathsf{x}_2 \cdot \mathsf{g}_\mathsf{p}^{-wt_1} \cdot \mathsf{h}^{-rt_1}) = \mathbf{e}(\mathsf{v}_2^{\mathsf{s}_1}, \mathsf{g}_\mathsf{p}^{-wt_1} \cdot \mathsf{h}^{-rt_1})
\end{aligned}$$

The IBE-IND-CPA and IBE-ANO properties still hold since the DBDH and DLIN assumptions still hold. We only need to prove the *poly*-KEY-ANO security.

Theorem 4. *The extended scheme Π' [Fig. 13] is poly-KEY-ANO secure under Definition 9 assuming CDDH is intractable.*

Proof. First, let us show an important propriety 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 $\mathsf{R}_0, \mathsf{R}_1, \mathsf{R}_2 \in \mathbb{G}_{\mathsf{q}}$:

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

Let $\mathcal{A}$ be any PPT adversary against *poly*-KEY-ANO$_{\Pi',\mathcal{A}}$. Using a hybrid argument, we can now drastically simplify the security model [Fig. 7], so that it looks like the one presented in Fig. 14, which we call 5-KEY-ANO. We reiterate that it is enough to show that Π' is secure according to model 5-KEY-ANO only because from two keys honestly extracted from the same identity we can generate a third with fresh random coins, without knowing the master secret-key.

In *poly*-KEY-ANO$_{\Pi',\mathcal{A}}$, $\mathcal{A}$ submits two lists list_0 and list_1 of the same length, say L, for the challenge. Since $\mathcal{A}$ is polynomially bounded, so is the length of the lists. L intermediate lists are constructed, such that $\mathsf{list}^0 = \mathsf{list}_0$ and for every $\mathsf{i} \in \{1..\mathsf{L}\}$, $\mathsf{list}^{\mathsf{i}} = \mathsf{list}^{\mathsf{i}-1}$, except for the element $\mathsf{list}^{\mathsf{i}}[\mathsf{i}]$ which is taken from $\mathsf{list}_1[\mathsf{i}]$. This results in $\mathsf{list}^{\mathsf{L}}$ being equal to 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 $\mathsf{i} \in \{0..\mathsf{L}\}$. 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-ANO. 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-ANO into a L-tuple of keys that corresponds to the requirements of eitheir $\mathsf{list}^{\mathsf{i}-1}$ 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-ANO 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.

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

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

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

Fig. 14. 5-KEY-ANO$_{\Pi,\mathcal{A}}$ Game

The model can be further simplified to that of Fig. 15, which we call 4-KEY-ANO. Again, making use of the property introduced in the beginning of this proof, 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 equivalente to that of distinguishuing 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 $(\mathsf{id}_0, \mathsf{id}_0, \mathsf{id}_0, \mathsf{id}_0)$. This also applies for the 5-tuple of keys extracted from $(\mathsf{id}_0, \mathsf{id}_0, \mathsf{id}_1, \mathsf{id}_1, \mathsf{id}_1)$. So, the advantage $\mathcal{A}$ has in distinguishing the tuples in 5-KEY-ANO game cannot be more than twice the advantage $\mathcal{A}$ has in distinguishing the tuples in 4-KEY-ANO.

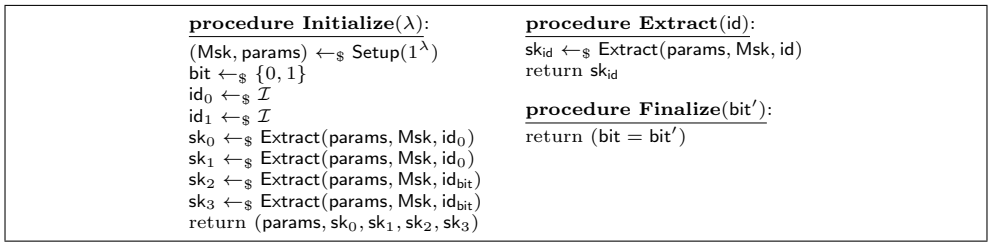

Fig. 15. 4-KEY-ANO$_{\Pi,\mathcal{A}}$ Game

To complete the proof, we build a simulator $\mathcal{S}_3$ [Fig. 16] that by playing game CDDH 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-ANO$_{\Pi',\mathcal{A}}$ can be forward to game CDDH. 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 $\mathsf{u} \in \mathbb{Z}_\mathsf{n}$.

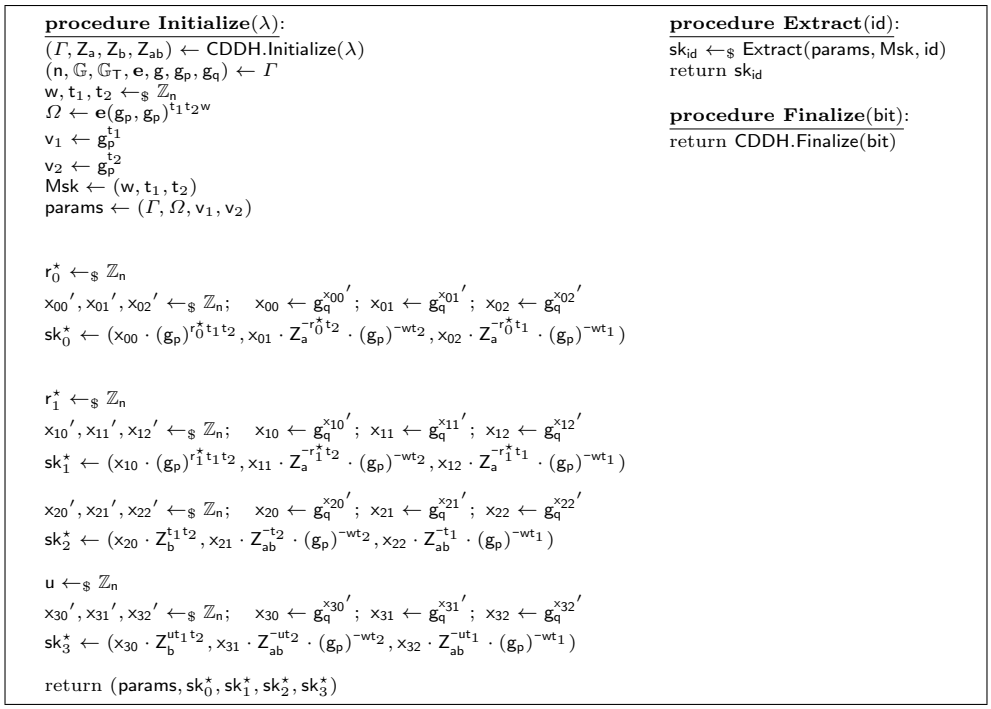

Fig. 16. Simulator $\mathcal{S}_3$ forwards $\mathcal{A}$'s guess from 4-KEY-ANO$_{\Pi',\mathcal{A}}$ to game CDDH.

Finally, we have that $\mathbf{Adv}^{poly\text{-KEY-ANO}}_{\Pi',\mathcal{A}}(\lambda) \leq 2\mathsf{L} \cdot \mathbf{Adv}_{\mathsf{CDDH},\mathcal{S}_3}$, which concludes our proof. □

6 Conclusion

We investigated the trapdoor privacy issues in ASE schemes and presented two privacy definitions. We also proposed two key anonymity properties (i.e. 2-KEY-ANO and poly-KEY-ANO) for IBE schemes so that these properties directly lead to the trapdoor privacy properties in the generic transformation from IBE to ASE [8]. We then proved that a simplified Boyen-Waters scheme achieves the 2-KEY-ANO security in the random oracle model and an extended version of this simplified scheme based on pairings over composite-order groups achieves poly-KEY-ANO security without random oracles. Our work leaves a number of interesting future research topics. One is

to enhance the proposed definitions by considering non-uniform message distributions in the challenge, as remarked in Section 3.4. Another is to analyze the trapdoor privacy properties of the schemes from [14, 20, 22], and have new constructions without random oracles. Another is to investigate the trapdoor privacy properties for ASE schemes that support search queries with more complex comparison structures. Yet another is to formally investigate the one-wayness property in the second scenario, namely the trapdoor privacy in the case that trapdoors match with ciphertexts.

Note.

After submitting this paper, we found that another paper by Boneh, Raghunathan, and Segev (http://eprint.iacr.org/2013/283) has addressed a similar problem. We will compare our results with theirs and update the paper soon.

References

1. Tang, Q.: Search in Encrypted Data: Theoretical Models and Practical Applications. In: Theory and Practice of Cryptography Solutions for Secure Information Systems. IGI (2013) 84–108
2. Song, D.X., Wagner, D., Perrig, A.: Practical Techniques for Searches on Encrypted Data. In: IEEE Symposium on Security and Privacy. (2000) 44–55
3. Boneh, D., Crescenzo, G.D., Ostrovsky, R., Persiano, G.: Public Key Encryption with Keyword Search. In: Advances in Cryptology — EUROCRYPT 2004. (2004) 506–522
4. Shen, E., Shi, E., Waters, B.: Predicate privacy in encryption systems. In: Proceedings of the 6th Theory of Cryptography Conference on Theory of Cryptography. (2009) 457–473
5. Fuhr, T., Paillier, P.: Decryptable searchable encryption. In: Proceedings of the 1st international conference on Provable security. (2007) 228–236
6. Hofheinz, D., Weinreb, E.: Correlation-resistant storage via keyword-searchable encryption. Cryptology ePrint Archive: Report 2008/423 (2008)
7. Tang, Q., Chen, X.: Towards asymmetric searchable encryption with message recovery and flexible search authorization. In: 8th ACM Symposium on Information, Computer and Communications Security (ASIACCS 2013). (2013) To appear
8. 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. J. Cryptol. **21**(3) (2008) 350–391
9. Kiltz, E.: From selective-id to full security: The case of the inversion-based boneh-boyen ibe scheme. Cryptology ePrint Archive: Report 2007/033 (2007)
10. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Advances in Cryptology – EUROCRYPT 2006. (2006) 409–426
11. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. In: Advances in Cryptology – CRYPTO 2001. (2001) 213–229
12. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Advances in Cryptology – CRYPTO 2004. (2004) 41–55
13. Boneh, D., Goh, E.J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Theory of Cryptography, Second Theory of Cryptography Conference (TCC). (2005) 325–341
14. Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. In: Proceedings of the 4th conference on Theory of cryptography. (2007) 535–554
15. Bellare, M., Boldyreva, A., Micali, S.: Public-key encryption in a multi-user setting: security proofs and improvements. In: Advances in Cryptology – EUROCRYPT 2000. (2000) 259–274
16. Boneh, D., Boyen, X.: Secure identity based encryption without random oracles. In: Advances in Cryptology – CRYPTO 2004. (2004) 443–459
17. Boneh, D., Boyen, X.: Efficient selective-id secure identity-based encryption without random oracles. In: Advances in Cryptology – EUROCRYPT 2004. (2004) 223–238
18. Waters, B.: Efficient identity-based encryption without random oracles. In: Advances in Cryptology – EUROCRYPT 2005. (2005) 114–127

19. Gentry, C.: Practical identity-based encryption without random oracles. In: Advances in Cryptology – EUROCRYPT 2006. (2006) 445–464
20. Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (without random oracles). In: Advances in Cryptology – CRYPTO 2006. (2006) 290–307
21. Attrapadung, N., Furukawa, J., Gomi, T., Hanaoka, G., Imai, H., Zhang, R.: Efficient identity-based encryption with tight security reduction. In: Proceedings of the 5th international conference on Cryptology and Network Security. (2006) 19–36
22. Ducas, L.: Anonymity from asymmetry: new constructions for anonymous hibe. In: Proceedings of the 2010 international conference on Topics in Cryptology (CT-RSA'10). (2010) 148–164
23. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A concrete security treatment of symmetric encryption. In: Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS). (1997) 394–403

Appendix I: Property Definitions for IBE

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

Definition 10. *An IBE scheme Π is IBE-IND-CPA secure if, for every legitimate PPT adversary $\mathcal{A}$, the following definition of advantage is negligible in λ*

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

where game $\mathsf{IBE\text{-}IND\text{-}CPA}_{\Pi,\mathcal{A}}$ is described in Fig. 17. The adversary is legitimate if it only calls Real-or-Random once.

procedure Initialize(λ):	**procedure Real-or-Random**$(\mathsf{id}^\star, \mathsf{m}_0)$:
$(\mathsf{Msk}, \mathsf{params}) \leftarrow_\$ \mathsf{Setup}(\lambda)$	if $\mathsf{id}^\star \in$ list return $\perp$
$\mathsf{bit} \leftarrow_\$ \{0,1\}$	$\mathsf{m}_1 \leftarrow_\$ \mathcal{M}$
return params	$\mathsf{c} \leftarrow_\$ \mathsf{Enc}(\mathsf{params}, \mathsf{m}_{\mathsf{bit}}, \mathsf{id}^\star)$
	return c
procedure Extract(id):	**procedure Finalize**(bit'):
if $\mathsf{id} == \mathsf{id}^\star$ return $\perp$	return $(\mathsf{bit} = \mathsf{bit}')$
$\mathsf{sk}_{\mathsf{id}} \leftarrow_\$ \mathsf{Extract}(\mathsf{params}, \mathsf{Msk}, \mathsf{id})$	
list $\leftarrow \mathsf{id} :$ list	
return $\mathsf{sk}_{\mathsf{id}}$	

Fig. 17. $\mathsf{IBE\text{-}IND\text{-}CPA}_{\Pi,\mathcal{A}}$ Game

Definition 11. *An IBE scheme Π is IBE-ANO secure if, for every legitimate PPT adversary $\mathcal{A}$, the following definition of advantage is negligible in λ*

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

where game $\mathsf{ANO}_{\Pi,\mathcal{A}}$ is described in Fig. 18. The adversary is legitimate if it only calls Real-or-Random once.

procedure Initialize(λ):	**procedure Real-or-Random**$(\mathsf{id}_0, \mathsf{m}^\star)$:
$(\mathsf{Msk}, \mathsf{params}) \leftarrow_\$ \mathsf{Setup}(\lambda)$	if $\mathsf{id}_0 \in$ list return $\perp$
$\mathsf{bit} \leftarrow_\$ \{0,1\}$	$\mathsf{id}_1 \leftarrow_\$ \mathcal{I}$
return params	$\mathsf{c} \leftarrow_\$ \mathsf{Enc}(\mathsf{params}, \mathsf{m}^\star, \mathsf{id}_{\mathsf{bit}})$
	$\mathsf{id}^\star \leftarrow \mathsf{id}_0$
	return c
procedure Extract(id):	**procedure Finalize**(bit'):
if $\mathsf{id} == \mathsf{id}^\star$ return $\perp$	return $(\mathsf{bit} = \mathsf{bit}')$
$\mathsf{sk}_{\mathsf{id}} \leftarrow_\$ \mathsf{Extract}(\mathsf{params}, \mathsf{Msk}, \mathsf{id})$	
list $\leftarrow \mathsf{id} :$ list	
return $\mathsf{sk}_{\mathsf{id}}$	

Fig. 18. $\mathsf{IBE\text{-}ANO}_{\Pi,\mathcal{A}}$ Game

Remark 4. Similar to the results in [23], we can show that the Real-or-Random challenge and the typical Left-or-Right style challenge are equivalent in asymmetric encryption security definitions when the message space is not polynomial size.

Appendix II: Property Definitions for ASE

Definition 12. *An ASE scheme $\mathcal{E}$ is computationally consistent if, for every legitimate PPT adversary $\mathcal{A}$, the following definition of advantage is negligible in λ*

$$\mathbf{Adv}^{\mathsf{ASE\text{-}CONSIST}}_{\mathcal{E},\mathcal{A}}(\lambda) := 2 \cdot \Pr[\mathsf{ASE\text{-}CONSIST}_{\mathcal{E},\mathcal{A}}(\lambda) \Rightarrow \mathsf{True}] - 1,$$

where game $\mathsf{ASE\text{-}CONSIST}_{\mathcal{E},\mathcal{A}}$ *is described in Fig. 19.*

procedure Initialize(λ):	**procedure Finalize**$(\mathsf{w}_0, \mathsf{w}_1)$:
$(\mathsf{sk}, \mathsf{pk}) \leftarrow_{\$} \mathsf{KeyGen}(\lambda)$	$\mathsf{c} \leftarrow_{\$} \mathsf{PEKS}(\mathsf{pk}, \mathsf{w}_0)$
return pk	$\mathsf{tp} \leftarrow_{\$} \mathsf{Trapdoor}(\mathsf{sk}, \mathsf{w}_1)$
	return $\mathsf{Test}(\mathsf{pk}, \mathsf{c}, \mathsf{tp}) \wedge (\mathsf{w}_0 \neq \mathsf{w}_1)$

Fig. 19. ASE-CONSIST Game

Definition 13. *An ASE scheme $\mathcal{E}$ is ASE-IND-CPA secure if, for every legitimate PPT adversary $\mathcal{A}$, the following definition of advantage is negligible in λ*

$$\mathbf{Adv}^{\mathsf{ASE\text{-}IND\text{-}CPA}}_{\mathcal{E},\mathcal{A}}(\lambda) := 2 \cdot \Pr[\mathsf{ASE\text{-}IND\text{-}CPA}_{\mathcal{E},\mathcal{A}}(\lambda) \Rightarrow \mathsf{True}] - 1,$$

where game $\mathsf{ASE\text{-}IND\text{-}CPA}_{\mathcal{E},\mathcal{A}}$ *is described in Fig. 20. The adversary is legitimate if it only calls Real-or-Random once.*

procedure Initialize(λ):	**procedure Real-or-Random**(w_0):
$(\mathsf{sk}, \mathsf{pk}) \leftarrow_{\$} \mathsf{KeyGen}(\lambda)$	if $\mathsf{w}_0 \in \mathsf{list}$ return $\perp$
$\mathsf{bit} \leftarrow_{\$} \{0,1\}$	$\mathsf{w}_1 \leftarrow_{\$} \mathcal{M}'$
return pk	$\mathsf{c} \leftarrow_{\$} \mathsf{PEKS}(\mathsf{pk}, \mathsf{w}_{\mathsf{bit}})$
	return c
procedure Trapdoor(w):	**procedure Finalize**(bit'):
if $\mathsf{w} == \mathsf{w}_0$ return $\perp$	return $(\mathsf{bit} = \mathsf{bit}')$
$\mathsf{tp} \leftarrow_{\$} \mathsf{Trapdoor}(\mathsf{sk}, \mathsf{w})$	
$\mathsf{list} \leftarrow \mathsf{w} : \mathsf{list}$	
return tp	

Fig. 20. $\mathsf{ASE\text{-}IND\text{-}CPA}_{\mathcal{E},\mathcal{A}}$ Game