Towards Asymmetric Searchable Encryption with Message Recovery and Flexible Search Authorization

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

Xiaofeng Chen
State Key Laboratory of Integrated Service Networks (ISN)
Xidian University, Xian 710071, P.R. China
xfchen@xidian.edu.cn

ABSTRACT

When outsourcing data to third-party servers, searchable encryption is an important enabling technique which simultaneously allows the data owner to keep his data in encrypted form and the third-party servers to search in the ciphertexts. Motivated by an *encrypted email retrieval and archive* scenario, we investigate asymmetric searchable encryption (ASE) schemes which support two special features, namely message recovery and flexible search authorization. With this new primitive, a data owner can keep his data encrypted under his public key and assign different search privileges to third-party servers. In the security model, we define the standard IND-CCA security against any outside attacker and define adapted ciphertext indistinguishability properties against inside attackers according to their functionalities. Moreover, we take into account the potential information leakage from trapdoors, and define two trapdoor security properties. Employing the bilinear property of pairings and a deliberately-designed double encryption technique, we present a provably secure instantiation of the primitive based on the DLIN and BDH assumptions in the random oracle model.

Categories and Subject Descriptors

E.3 [**Data Encryption**]: Public Key Cryptosystems

Keywords

Cloud Computing; Data Outsourcing; Searchable Encryption; Privacy

1. INTRODUCTION

To protect outsourced data and services in the cloud computing environment, cryptographic researchers have devoted a lot of efforts to searchable encryption techniques. Such techniques are particularly interesting because they allow a data owner to encrypt his data and outsource the cipheretxts while still being able to let third-party service providers search on his behalf without leaking any unnecessary information. Roughly speaking, searchable encryption schemes fall into two categories. One category is symmetric searchable encryption (SSE) schemes, represented by the work of Song, Wagner, and Perrig [12]. In this category, only the data owner can contribute searchable contents. The other category is asymmetric searchable encryption (ASE) schemes, represented by the work of Boneh et al. [4]. In this category, the concept of public key encryption is employed so that every entity can contribute searchable contents. A detailed survey and analysis of existing searchable encryption (both SSE and ASE) schemes can be found in [13].

Motivated by an *encrypted email retrieval and archive* scenario, as described below, we investigate ASE schemes which support two special features: *message recovery* and *flexible search authorization*. The message recovery feature requires that a ciphertext not only allows the data owner to recover the plaintext but also allows third-party servers to search in it. The flexible searchable authorization feature requires that the data owner can authorize a third-party server in three different ways: (1) authorize the server to search any message at the data owner's interest by assigning a message-dependent trapdoor (i.e. the server can only determine whether the message encoded in the trapdoor is equal to the plaintext inside a ciphertext); (2) authorize the server to search any message at the server's interests by assigning a master trapdoor (i.e. the server can choose a message at its will and see whether it is equal to the plaintext inside any ciphertext); (3) authorize the server to perform both types of searches. Throughout the paper, we refer to this new type of ASE schemes as ASE††.

1.1 Encrypted Email Retrieval and Archive

Suppose that Bob is an employee of the company COM and emails sent to him are required to be encrypted and stored in an email server managed by COM. Suppose that Alice wants to send an email to Bob, then she can encrypt the email using Bob's public key and send the ciphertext to COM's email server. Note that, here, Bob is the owner of his emails. In practice, the underlying encryption scheme should satisfy the following requirements.

1. When Bob is traveling around, he may want to selectively retrieve and read his emails from COM's email server. Thus, the encryption scheme should allow the email server to search on Bob's behalf to identify those at his interests.

2. A malicious user can send Bob encrypted emails, which contain malwares or viruses. Thus, the encryption

ASIA CCS'13, May 8–10, 2013, Hangzhou, China.

scheme should allow the email server to scan the encrypted emails to identify malicious contents.

3. Bob may change his job over the time, so that he may want to archive his emails during different jobs in a cloud server, such as Gmail. Bob can simply forward all his encrypted emails to the archive server. Later on, Bob may need to selectively retrieve some of the emails, therefore, the encryption scheme should allow the cloud server to search on Bob's behalf as in the first requirement.

With an ASE†† scheme, the message recovery feature guarantees that only Bob can decrypt the encrypted emails while he can still authorize the servers to search on his behalf. The flexible search authorization feature allows Bob to assign a master trapdoor to COM's email server so that the latter can scan malicious contents inside the encrypted emails, and this feature also allows Bob to assign message-dependent trapdoors to COM's email server and the cloud server to search emails at his interests.

1.2 Related Work

As surveyed in [13], the majority of existing ASE schemes are index-based, which means that they only aim at supporting search over scrambled keywords and typically do not allow the data owner to recover the keywords. By definition, these schemes do not allow the servers to search directly over the contents, therefore their functionality is far from what an ASE†† scheme is aimed for. On the other hand, an ASE†† scheme fulfills the purpose of these index-based ASE schemes.

Fuhr and Paillier [7] and Hofheinz and Weinreb [8] investigated the concept of ASE with message recovery. Their formulations only allow the data owner to assign message-dependent trapdoors to third-party servers, thus provide less functionality than ASE††. As to the security models, the authors only consider information leakage from ciphertexts and allow the servers to easily recover the information encoded in the trapdoors. In [7], if a match is found then the server immediately knows the plaintext in the ciphertext, while, in [8], the to-be-searched message is sent to the server in plaintext. In practice, this may be regarded as a serious security weakness.

Ibraimi et al. [9] pushed forward the concept of ASE with message recovery and proposed a new primitive PKEDS, namely public key encryption with delegated search. With a PKEDS scheme, a data owner can authorize third-party servers in two ways: (1) authorize a server to search any message at the server's interests by assigning a master trapdoor; (2) authorize a server to search messages at the data owner's interests by assigning message-dependent trapdoors. In their formulation, authorization (2) implies authorization (1), because search based on message-dependent trapdoors also requires a master trapdoor as input. In other words, the data owner must assign a master trapdoor to a server in order to ask the latter to perform any search. This fact conflicts with the *least privilege* principle in information security and is undesirable. For instance, in the aforementioned application scenario, Bob may not want to assign a master trapdoor to the cloud server to let the latter probe all his emails.

Recently, Tang et al. in [14] refined PKEDS and proposed a primitive similar to ASE††. However, in their security model, only IND-CPA security is considered and the notion of soundness is also weaker than that in this paper.

1.3 Contribution

In this paper, we formulate a new primitive, namely ASE††. With an ASE†† scheme, the data owner can keep data in encrypted form while still be able to recover the plaintext and authorize third-party servers to search on his behalf. The authorization to a server is through assigning the appropriate trapdoors: message-dependent trapdoors, master trapdoors, or both types of trapdoors. In contrast to [9], a search based on a message-dependent trapdoor does not require a master trapdoor as input, and this implies a significant security improvement. In practice, the data owner can authorize different servers based on their trustworthiness.

With respect to the functionality of ASE††, we provide a comprehensive definition for the soundness property. The property guarantees that not only the encryption/decryption algorithms work well but also the decryption and the test algorithms are bilaterally consistent with each other. We present a fine-grained security model by considering four categories of attackers, including an outside attacker and three types of curious servers based on the trapdoors they receive. We define the standard IND-CCA security against an outside attacker, and define adapted ciphertext indistinguishability security properties against the curious servers. Moreover, we define two trapdoor security properties to model the possible information leakages from message-dependent trapdoors. This security model is stronger than that in [14].

The soundness property turns out to be very difficult to be satisfied. Hybrid constructions (e.g. [1, 16]) and other constructions (e.g. [14]) do not satisfy this property. Based on bilinear pairing techniques and a deliberately-designed double encryption technique, we propose a new ASE†† scheme and prove its security based on DLIN and BDH assumptions in the random oracle model.

1.4 Organization

In Section 2, we formulate the concept of ASE†† and define the soundness property. In Section 3, we present a fine-grained security model for ASE††. In Section 4, we present an IND-CCA secure scheme and analyse its security. In Section 5, we conclude the paper.

2. DEFINITION OF ASE††

Throughout the paper, we use the following notation. $x||y$ means the concatenation of x and y, P.P.T. means probabilistic polynomial time, $x \in_R X$ means that the element x is chosen from the set X uniformly at random, and $x \xleftarrow{\$} \mathcal{A}(m_1, m_2, \cdots; \mathcal{O}_1, \mathcal{O}_2, \cdots)$ means that x is the output of the algorithm $\mathcal{A}$ which runs with the input $m_1, m_2, \cdots$ and access to the oracles $\mathcal{O}_1, \mathcal{O}_2, \cdots$.

2.1 Primitive Formulation

In general, an ASE†† scheme involves the following entities. In each server category, there can be multiple servers, but we only consider one for the simplicity of description.

- A data owner, who is supposed to receive encrypted messages. This entity is also referred to as *receiver* in encryption schemes.
- Type-I server $\mathcal{S}_1$, which receives message-dependent trapdoors from the data owner. It can test whether the

message encoded in a trapdoor is equal to the plaintext inside any given ciphertext.

- Type-II server $\mathcal{S}_2$, which receives a master trapdoor from the data owner. It can choose a message at its will and test whether it is equal to the plaintext inside any given ciphertext.
- Hybrid server $\mathcal{S}_h$, which is both Type-I and Type-II.
- Senders, who may send messages to the data owner.

Let λ be the security parameter. Formally, an ASE†† scheme consists of the following algorithms.

- $\mathsf{rKeyGen}(\lambda)$: Run by the data owner, it outputs a public/private key pair (PK_r, SK_r). Let the message space be denoted as $\mathcal{W}$.
- $\mathsf{Encrypt}(w, PK_r)$ Run by a message sender, it outputs a ciphertext c_w for a message $w \in \mathcal{W}$.
- $\mathsf{Decrypt}(c_w, SK_r)$: Run by the data owner, it outputs the plaintext w or an error message $\perp$.
- $\mathsf{sKeyGen}(\lambda)$: Run by a server ($\mathcal{S}_1$, $\mathcal{S}_2$, or $\mathcal{S}_h$), it outputs a public/private key pair (PK_s, SK_s)
- $\mathsf{TrapGen}_1(w, PK_s, SK_r)$: Run by the data owner, it generates a message-dependent trapdoor $t_{w,s}$ for the server with public key PK_s.
- $\mathsf{Test}_1(c_w, t_{w',s}, SK_s)$: Run by the server with message-dependent trapdoor $t_{w',s}$ and private key SK_s, it returns 1 if $w' = w$ and 0 otherwise.
- $\mathsf{TrapGen}_2(PK_s, SK_r)$: Run by the data owner, it outputs a master trapdoor $t_{*,s}$ for the server with public key PK_s.
- $\mathsf{Test}_2(c_w, w', t_{*,s}, SK_s)$: Run by the server with the master trapdoor $t_{*,s}$ and private key SK_s, it returns 1 if $w' = w$ and 0 otherwise.

Note that ($\mathsf{rKeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt}$) define a standard PKE scheme. As indicated in [4], the trapdoors should be transmitted to the servers through a secure channel (with confidentiality), otherwise any attacker will be able to obtain the trapdoors and search over the ciphertexts. However, this issue has not been formally addressed in the primitive formulation and security model in [4]. Here, we explicitly provide the $\mathsf{sKeyGen}$ algorithm so that potential servers can run this algorithm to generate a key pair, with which the data owner can generate trapdoors in an encrypted form by running the $\mathsf{TrapGen}_1$ and $\mathsf{TrapGen}_2$ algorithms.

2.2 Soundness Property

Similar to the case for other primitives, the first property we want is soundness, defined as follows.

DEFINITION 1. *An ASE†† scheme is sound if, for any $(PK_r, SK_r) = \mathsf{rKeyGen}(\lambda)$ and $(PK_s, SK_s) = \mathsf{sKeyGen}(\lambda)$, the following conditions are satisfied.*

1. *For any $w \in \mathcal{W}$, $\mathsf{Decrypt}(\mathsf{Encrypt}(w, PK_r), SK_r) = w$ always holds.*
2. *For any ciphertext c, $\mathsf{Decrypt}(c, SK_r) = w$ if and only if $\mathsf{Test}_1(c, t_{w,s}, SK_s) = 1$.*
3. *For any ciphertext c, $\mathsf{Decrypt}(c, SK_r) = w$ if and only if $\mathsf{Test}_2(c, w, t_{*,s}, SK_s) = 1$.*

In the above definition, the first condition means that the encryption/decryption functionality works well. The second and the third conditions define the bilateral consistency property between the decryption and test algorithms.

The "if" condition guarantees that any matched ciphertext by the test algorithms can be successfully decrypted and the resulted plaintext will be equal to that assumed in the test algorithms. For instance, if a test algorithm indicates c is an encryption of w, then the decryption algorithm will not output $w' \neq w$ or an error $\perp$. Basically, this property guarantees that there is no "false positive" in the search process.

The "only if" condition guarantees that if a ciphertext can be successfully decrypted then the test algorithms should be able to properly match it. This property guarantees that no targeted ciphertext will be missed by the test algorithms. Take the *encrypted email retrieval and archive* scenario as example, this property is crucial for the email server not to miss any malicious contents in the encrypted emails.

3. THE SECURITY MODEL

We assume that the message senders possess a valid copy of the receiver's public key and the receiver possesses valid copies of the public keys of the servers. How to securely distribute these public keys should follow some standard practice, and we skip the discussion in this paper.

As to the security of an ASE†† scheme, there are two main privacy concerns.

- One concern is the leakage of plaintext information from ciphertexts, which is a standard concern for all encryption schemes. Given ciphertexts, an attacker can try to deduce information about the encrypted plaintexts. Particularly, for an ASE†† scheme, knowledge about the (un-encrypted) master trapdoors or message-dependent trapdoors will provide additional advantage to the attacker. Therefore, we will consider the following types of attackers.
  - Outside attacker: This type of attacker is not assigned with any type of (unencrypted) trapdoors.
  - Curious Type-I server $\mathcal{S}_1$: This type of attacker has been assigned with only message-dependent trapdoors generated under its public key.
  - Curious Type-II server $\mathcal{S}_2$: This type of attacker has only been assigned with a master trapdoor generated under its public key.
  - Curious hybrid server $\mathcal{S}_h$: This type of attacker has been assigned with a master trapdoor and message-dependent trapdoors generated under its public key.

  It is clear that a hybrid server is more powerful than others. However, due to the fact that the data owner may employ all three types of servers, it is necessary to consider the maximal level of security against each of them independently.
- The other concern is information leakage from message-dependent trapdoors. For example, the Type-I server $\mathcal{S}_1$ receives message-dependent trapdoors so that it

knows which ciphertext matches a received trapdoor. However, $\mathcal{S}_1$ should not know the message encoded in the trapdoor, or equivalently, $\mathcal{S}_1$ should not know what is the plaintext of the matched ciphertext. Furthermore, any entity other than $\mathcal{S}_1$ should not learn anything about the trapdoor. The same security requirement applies to a hybrid server $\mathcal{S}_h$ because it also gets message-dependent trapdoors from the receiver.

To facilitate our security definitions, we first detail all the potential oracles accessible to an attacker. Based on our assumption that there is only one server of every type, so that each key generation oracle ($\mathsf{sKeyGen}_1$, $\mathsf{sKeyGen}_2$, $\mathsf{sKeyGen}_h$) can only be queried once. Trivially, the key request oracles can only be queried after the corresponding key generation oracles have been queried.

- $\mathsf{sKeyGen}_1$: The challenger runs the $\mathsf{sKeyGen}$ algorithm to generate (PK_{s_1}, SK_{s_1}) for the Type-I server $\mathcal{S}_1$, and returns PK_{s_1}.
- $\mathsf{sKeyReq}_1$: The challenger returns SK_{s_1}.
- $\mathsf{sKeyGen}_2$: The challenger runs the $\mathsf{sKeyGen}$ algorithm to generate (PK_{s_2}, SK_{s_2}) for the Type-II server $\mathcal{S}_2$, and returns PK_{s_2}.
- $\mathsf{sKeyReq}_2$: The challenger returns SK_{s_2}.
- $\mathsf{sKeyGen}_h$: The challenger runs the $\mathsf{sKeyGen}$ algorithm to generate (PK_{s_h}, SK_{s_h}) for the hybrid server $\mathcal{S}_h$, and returns PK_{s_h}.
- $\mathsf{sKeyReq}_h$: The challenger returns SK_{s_h}.
- $\mathsf{TrapGen}_1$ query with a message w and PK_s as input: The challenger returns $\mathsf{TrapGen}_1(w, PK_s, SK_r)$. In this case s can be either s_1 or s_h
- $\mathsf{TrapGen}_2$ query with a public key PK_s: The challenger returns $\mathsf{TrapGen}_2(PK_s, SK_r)$. In this case s can be either s_2 or s_h
- $\mathsf{rKeyReq}_r$: The challenger returns the receiver's private key SK_r.
- $\mathsf{Decrypt}$ query with input c: The challenger returns $\mathsf{Decrypt}(c, SK_r)$.

In the following, we present security definitions with respect to the aforementioned two types of privacy concerns.

3.1 Ciphertext Security Properties

Against an outside attacker, in Definition 2, we define an IND-CCA property similar to the IND-CCA security for PKE. Informally, the property says that, given some ciphertexts and various oracle accesses (including decryption oracle), an attacker cannot learn anything about the plaintexts.

DEFINITION 2. *An ASE†† scheme is IND-CCA secure against an outside attacker if the attacker's advantage (i.e. $|Pr[b' = b] - \frac{1}{2}|$) is negligible in the game shown in Figure 1.*

In the attack game, $w_0, w_1 \in \mathcal{W}$ and *state* is some state information generated by the attacker. In Phase 4, the attacker is not allowed to query the $\mathsf{Decrypt}$ oracle with c_{w_b}. In this game, an outside attacker is modeled since it is not allowed to query any server's private key through $\mathsf{sKeyReq}_1$, $\mathsf{sKeyReq}_2$, or $\mathsf{sKeyReq}_h$. In practice, an outside attacker can eavesdrop on the transmission of (encrypted) trapdoors, so that we offer it access to both trapdoor generation oracles with its own inputs.

1. $(PK_r, SK_r) \xleftarrow{\$} \mathsf{rKeyGen}(\lambda)$
2. $(w_0, w_1, state) \xleftarrow{\$} \mathcal{A}(PK_r; \mathsf{sKeyGen}_1, \mathsf{sKeyGen}_2, \mathsf{sKeyGen}_h, \mathsf{TrapGen}_1, \mathsf{TrapGen}_2, \mathsf{Decrypt})$
3. $b \in_R \{0,1\},\ c_{w_b} = \mathsf{Encrypt}(w_b, PK_r)$
4. $b' \xleftarrow{\$} \mathcal{A}(PK_r, state, c_{w_b}; \mathsf{sKeyGen}_1, \mathsf{sKeyGen}_2, \mathsf{sKeyGen}_h, \mathsf{TrapGen}_1, \mathsf{TrapGen}_2, \mathsf{Decrypt})$

Figure 1: IND-CCA Security

REMARK 1. *To define the IND-CPA security against an outside attacker, we only need to disable the* $\mathsf{Decrypt}$ *oracle in the game shown in Figure 1. In fact, if an ASE†† scheme is secure under Definition 3 and Definition 7 (given below), then it is automatically IND-CPA secure. Due to the limit of space, we put the proof of this result in the full version of this paper.*

Against a curious Type-I server $\mathcal{S}_1$, in Definition 3, we define a ciphertext indistinguishability (CI) security property similar to the security definition of PEKS [4]. Informally, this property says that, given some ciphertexts *for which* $\mathcal{S}_1$ *has not obtained the corresponding message-dependent trapdoors*, then it cannot learn anything about the plaintexts.

DEFINITION 3. *An ASE†† scheme achieves CI security against a curious Type-I server $\mathcal{S}_1$, if the attacker's advantage (i.e. $|\Pr[b' = b] - \frac{1}{2}|$) is negligible in the game shown in Figure 2.*

1. $(PK_r, SK_r) \xleftarrow{\$} \mathsf{rKeyGen}(\lambda)$
2. $(w_0, w_1, state) \xleftarrow{\$} \mathcal{A}(PK_r; \mathsf{sKeyGen}_1, \mathsf{sKeyGen}_2, \mathsf{sKeyGen}_h, \mathsf{TrapGen}_1, \mathsf{TrapGen}_2, \mathsf{sKeyReq}_1)$
3. $b \in_R \{0,1\},\ c_{w_b} = \mathsf{Encrypt}(w_b, PK_r)$
4. $b' \xleftarrow{\$} \mathcal{A}(PK_r, state, c_{w_b}; \mathsf{sKeyGen}_1, \mathsf{sKeyGen}_2, \mathsf{sKeyGen}_h, \mathsf{TrapGen}_1, \mathsf{TrapGen}_2, \mathsf{sKeyReq}_1)$

Figure 2: CI Security against a Type-I Server

In the attack game, $w_0, w_1 \in \mathcal{W}$ and *state* is some state information generated by the attacker. In step 2 and 4, the $\mathsf{TrapGen}_1$ oracle should have not been queried with (w_0, PK_{s_1}) or (w_1, PK_{s_1}). In this game, a Type-I server is modeled since the attacker is allowed to obtain SK_{s_1} through a $\mathsf{sKeyReq}_1$ query, but has no access to SK_{s_2} or SK_{s_h}. The oracle access to message-dependent trapdoors for any messages except for w_0 and w_1 is indeed a big privilege for the attacker. Most existing IND-CPA secure PKE schemes, such as ElGamal [6], normally allow some sort of homomorphism, so that they will not be secure if directly used in constructing an ASE†† scheme. As an example, it is easy to verify that the PKEDS scheme in [9] is not secure under this definition.

Against a curious hybrid server $\mathcal{S}_h$, we define a special CI security property. Compared with Definition 2 and Definition 3, the speciality lies in that the challenger randomly chooses the challenge messages w_0, w_1 instead of letting the attacker do. The rationale is that, for a curious hybrid server $\mathcal{S}_h$, if it can choose the messages then it can easily tell which message is encrypted by running the Test_2 algorithm.

DEFINITION 4. *An ASE†† scheme achieves CI security against a curious hybrid server $\mathcal{S}_h$, if the attacker's advantage (i.e. $|\Pr[b' = b] - \frac{1}{2}|$) is negligible in the game shown in Figure 3.*

1. $(PK_r, SK_r) \xleftarrow{\$} \mathsf{rKeyGen}(\lambda)$
2. $state \xleftarrow{\$} \mathcal{A}(PK_r; \mathsf{sKeyGen}_1, \mathsf{sKeyGen}_2, \mathsf{sKeyGen}_h, \mathsf{TrapGen}_1, \mathsf{TrapGen}_2, \mathsf{sKeyReq}_h)$
3. $b \in_R \{0,1\}$, $w_0, w_1 \in_R \mathcal{W}$, $c_b = (\mathsf{Encrypt}(w_0, PK_r), \mathsf{Encrypt}(w_b, PK_r))$
4. $b' \xleftarrow{\$} \mathcal{A}(PK_r, state, c_b; \mathsf{sKeyGen}_1, \mathsf{sKeyGen}_2, \mathsf{sKeyGen}_h, \mathsf{TrapGen}_1, \mathsf{TrapGen}_2, \mathsf{sKeyReq}_h)$

Figure 3: CI Security against a Hybrid Server

In the attack game, *state* is some state information generated by the attacker. In this game, a hybrid server is modeled since the attacker is allowed to obtain SK_{s_h} through a $\mathsf{sKeyReq}_h$ query, but has no access to SK_{s_1} or SK_{s_2}. This property guarantees that, given some ciphertexts, the attacker can neither recover the plaintexts nor find the equality relationship of the plaintexts. Therefore, it provides much stronger security guarantee than the standard one-wayness property used in [9] and the enhanced one-wayness property by Bellare, Boldyreva and O'Neill [2].

Similarly, we can define a special CI security property against a curious Type-II server, as shown in Definition 5. It is clear that if an ASE†† scheme is secure under Definition 4 then it is also secure under Definition 5.

DEFINITION 5. *An ASE†† scheme achieves CI security against a curious Type-II server $\mathcal{S}_2$, if the attacker's advantage (i.e. $|\Pr[b' = b] - \frac{1}{2}|$) is negligible in the game shown in Figure 3, where the $\mathsf{sKeyReq}_h$ oracle is replaced with the $\mathsf{sKeyReq}_2$ oracle.*

REMARK 2. *It is worth noting that the security properties defined in Definition 4 and 5 only make sense when the message space of the encryption scheme is not polynomial size. when the message space is polynomial size, the attacker can perform brute-force attacks. This fact is also true for Definition 6.*

3.2 Trapdoor Security Properties

In Definition 6, we define the universal trapdoor one-wayness property. The property guarantees that no entity can recover the message encoded in a message-dependent trapdoor. Note that this will require the message space $\mathcal{W}$ not to be polynomial size. In the attack game, an outside attacker and various servers are modeled because the attacker can obtain the private keys $(SK_{s_1}, SK_{s_2}, SK_{s_h})$ through the key request oracles $(\mathsf{sKeyReq}_1, \mathsf{sKeyReq}_2, \mathsf{sKeyReq}_h)$.

DEFINITION 6. *An ASE†† scheme achieves universal trapdoor one-wayness if the attacker's advantage (i.e. $\Pr[w = w']$) is negligible in the game shown in Figure 4. Note that the output value t in Phase 2 can be either 1, or 2, or h.*

1. $(PK_r, SK_r) \xleftarrow{\$} \mathsf{rKeyGen}(\lambda)$
2. $(t, state) \xleftarrow{\$} \mathcal{A}(PK_r; \mathsf{sKeyGen}_1, \mathsf{sKeyGen}_2, \mathsf{sKeyGen}_h, \mathsf{TrapGen}_1, \mathsf{TrapGen}_2, \mathsf{sKeyReq}_1, \mathsf{sKeyReq}_2, \mathsf{sKeyReq}_h)$
3. $w \in_R \mathcal{W}$, $t_{w,s_t} = \mathsf{TrapGen}_1(w, PK_{s_t}, SK_r)$
4. $w' \xleftarrow{\$} \mathcal{A}(PK_r, state, t_{w,s_t}; \mathsf{sKeyGen}_1, \mathsf{sKeyGen}_2, \mathsf{sKeyGen}_h, \mathsf{TrapGen}_1, \mathsf{TrapGen}_2, \mathsf{sKeyReq}_1, \mathsf{sKeyReq}_2, \mathsf{sKeyReq}_h, \mathsf{rKeyReq}_r)$

Figure 4: Universal Trapdoor One-wayness

To further strengthen the universal trapdoor one-wayness property, in Definition 7, we define the *message-dependent trapdoor indistinguishability* property. Informally, the property guarantees that, for the Type-I server $\mathcal{S}_1$ or the hybrid server $\mathcal{S}_h$, no other entity can learn anything about the messages encoded in message-dependent trapdoors generated under its public key.

DEFINITION 7. *An ASE†† scheme achieves trapdoor indistinguishability if the attacker's advantage (i.e. $|\Pr[b' = b] - \frac{1}{2}|$) is negligible in the game shown in Figure 5.*

1. $(PK_r, SK_r) \xleftarrow{\$} \mathsf{rKeyGen}(\lambda)$
2. $(w_0, w_1, t, state) \xleftarrow{\$} \mathcal{A}(PK_r; \mathsf{sKeyGen}_1, \mathsf{sKeyGen}_2, \mathsf{sKeyGen}_h, \mathsf{TrapGen}_1, \mathsf{TrapGen}_2, \mathsf{sKeyReq}_{t'}, \mathsf{sKeyReq}_2)$
3. $b \in_R \{0,1\}$, $t_{w_b} = \mathsf{TrapGen}_1(w_b, PK_{s_t}, SK_r)$
4. $b' \xleftarrow{\$} \mathcal{A}(PK_r, state, t_{w_b}; \mathsf{sKeyGen}_1, \mathsf{sKeyGen}_2, \mathsf{sKeyGen}_h, \mathsf{TrapGen}_1, \mathsf{TrapGen}_2, \mathsf{sKeyReq}_{t'}, \mathsf{sKeyReq}_2, \mathsf{rKeyReq}_r)$

Figure 5: Trapdoor Indistinguishability

In the attack game, $w_0, w_1 \in \mathcal{W}$, the values t, t' are chosen (by the attacker) from $\{1, h\}$ and $t \neq t'$, and *state* is some state information generated by the attacker. This guarantees that $\mathsf{sKeyReq}_t$ is not queried in the game. We allow the attacker to request the receiver's private key in Phase 2 and all servers' private keys except for SK_{s_t}. It captures our intention that, even if the receiver's private key is leaked or compromised, then the attacker still cannot figure out what the receiver has searched for. This is similar to the forward secrecy property in key establishment protocols.

4. IND-CCA SECURE ASE††

In this section, we first briefly mention some attempts, and then describe a novel ASE†† scheme and analyse its security properties in our security model.

4.1 Some Attempts

To construct an ASE†† scheme, a very natural direction is to follow the hybrid method used in [16]. Generically, the data owner can have two key pairs (pk_1, sk_1) and (pk_2, sk_2),

and the ciphertext of a message m is in the form of $c = (c_1, c_2) = (\mathsf{Encrypt}_1(m, pk_1), \mathsf{Encrypt}_2(m, pk_2))$, where c_1 can be used for decryption and c_2 can be used for search. With this methodology, the difficulty lies in that we need to provide a knowledge proof that c_1 and c_2 contain the same message in order to achieve the soundness property. Moreover, how to generate master trapdoors and message-dependent trapdoors is not a straightforward task either. It remains as an open problem to have a generic construction for ASE[††].

Instead of a generic construction, we may follow the semi-hybrid method in [14]: use a standard PKE scheme as a main component and employ a key-private IBE scheme to facilitate trapdoor constructions. In order to achieve the soundness property, we still need to provide a knowledge proof to guarantee that the decryption and the test functions are consistent with each other (note that the scheme in [14] does not satisfy the third requirement in Definition 1). Moreover, it is also an interesting future work to improve this scheme to achieve IND-CCA security.

4.2 Proposed Scheme and its Soundness

In the proposed scheme, the intuition behind the $\mathsf{Encrypt}$ algorithm is that a message is protected by two layers of encryption (see the element c_2). The first layer of encryption makes use of a hash value of the message and can only be removed by the receiver, while the second layer makes use of a hash value of the message and a bilinear pairing technique and it can only be removed with an appropriate message-dependent trapdoor or a master trapdoor. The search algorithms (Test_1 and Test_2) are made possible by employing the bilinear property of pairings.

- $\mathsf{sKeyGen}(\lambda)$: It generates a key pair (PK_s, SK_s) for a standard PKE scheme $(\mathsf{KeyGen}', \mathsf{Encrypt}', \mathsf{Decrypt}')$, which has the message space $\mathcal{M}$.

- $\mathsf{rKeyGen}(\lambda)$: It generates (PK_r, SK_r) as follows.

  1. Generate the pairing parameters: group $\mathbb{G}$ of prime order p, a bilinear map $\hat{e}$: $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, randomly chosen generators g_1 and g_2 of $\mathbb{G}$.
  2. Select a symmetric key encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ whose key space is $\mathcal{M}$, and select three hash functions: $\mathsf{H}_1 : \mathbb{G}_T \to \mathbb{G}$, $\mathsf{H}_2 : \mathbb{G} \to \mathbb{G}$, and $\mathsf{H}_3 : \{0,1\}^* \to \{0,1\}^\lambda$.
  3. Select $x_1, x_2 \in_R \mathbb{Z}_p$ and generate the receiver's key pair (PK_r, SK_r) where $PK_r = (g_1^{x_1},\ g_2^{x_2})$ and $SK_r = (x_1,\ x_2)$.

  Besides PK_r, the pairing parameters $(\mathbb{G}, \mathbb{G}_T, \hat{e}, p, g_1, g_2)$, symmetric key encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$, and hash functions H_1, H_2, and H_3 should be made public. The receiver's message space $\mathcal{W}$ is $\mathbb{G}$.

- $\mathsf{Encrypt}(w, PK_r)$: It selects $r_1, r_2 \in_R \mathbb{Z}_p$ and generates a ciphertext $c_w = (c_1, c_2, c_3, c_4, c_5)$ as follows.

$$c_1 = g_1^{r_1},\ c_2 = \mathsf{H}_1(\hat{e}(g_1^{x_1}, \mathsf{H}_2(w)^{r_1})) \cdot \mathsf{H}_2(w)^{r_1} \cdot w,$$

$$c_3 = g_2^{r_2},\ c_4 = g_2^{x_2 r_2} \cdot \mathsf{H}_2(w)^{r_1},$$

$$c_5 = \mathsf{H}_3(c_1||c_2||c_3||c_4||\mathsf{H}_2(w)||\hat{e}(g_1^{x_1}, \mathsf{H}_2(w)^{r_1})).$$

- $\mathsf{Decrypt}(c_w, SK_r)$: It parses c_w as $(c_1, c_2, c_3, c_4, c_5)$ and computes $w = \frac{c_2}{\mathsf{H}_1(\hat{e}(g_1^{x_1}, \frac{c_4}{c_3^{x_2}})) \cdot \frac{c_4}{c_3^{x_2}}}$, and outputs w if the following equalities hold.

$$\hat{e}(c_1, \mathsf{H}_2(w)) = \hat{e}(g_1, \frac{c_4}{c_3^{x_2}}))$$

$$c_5 = \mathsf{H}_3(c_1||c_2||c_3||c_4||\mathsf{H}_2(w)||\hat{e}(g_1^{x_1}, \frac{c_4}{c_3^{x_2}}))$$

  If either equality does not hold, the algorithm outputs an error symbol $\perp$.

- $\mathsf{TrapGen}_1(w, PK_s, SK_r)$: It selects $y \in_R \mathbb{Z}_p$ and computes the message-dependent trapdoor $t_{w,s} = (v_7, v_8)$ as follows.

$$v_0 = \mathsf{H}_2(w)^{x_1 y},\ v_1 = \mathsf{H}_2(w)^{x_1},\ v_2 = \hat{e}(g_1^{x_1 y}, w^{-1}),$$

$$v_3 = g_1^{x_1 y},\ v_4 = g_1^{x_1 x_2 y},\ v_5 = \mathsf{H}_2(w),$$

$$k_1 \in_R \mathcal{M},\ v_7 = \mathsf{Enc}(v_0||v_1||v_2||v_3||v_4||v_5, k_1),$$

$$v_8 = \mathsf{Encrypt}'(k_1, PK_s).$$

- $\mathsf{Test}_1(c, t_{w,s}, SK_s)$: It performs as follows.

  1. Parse c as $(c_1', c_2', c_3', c_4', c_5')$ and $t_{w,s}$ as (v_7, v_8).
  2. Decrypt v_8 to obtain k_1 and decrypt v_7 to recover $v_0, v_1, v_2, v_3, v_4, v_5$.
  3. Output 1 if the following equalities hold, and output 0 otherwise.

$$\frac{\hat{e}(v_3, c_4')}{\hat{e}(v_4, c_3')} = v_2 \cdot \hat{e}(v_3, \frac{c_2'}{\mathsf{H}_1(\hat{e}(c_1', v_1))}) = \hat{e}(c_1', v_0) \quad (1)$$

$$c_5' = \mathsf{H}_3(c_1'||c_2'||c_3'||c_4'||v_5||\hat{e}(c_1', v_1)) \quad (2)$$

- $\mathsf{TrapGen}_2(PK_s, SK_r)$: It generates a master trapdoor $t_{*,s} = (u_1, u_2)$, where $z \in_R \mathbb{Z}_p$, $k_2 \in_R \mathcal{M}$,

$$u_1 = \mathsf{Enc}(x_1||g_1^z||g_1^{x_2 \cdot z}, k_2),\ u_2 = \mathsf{Encrypt}'(k_2, PK_s).$$

- $\mathsf{Test}_2(c, w, t_{*,s}, SK_s)$: It performs as follows.

  1. Parse c as $(c_1', c_2', c_3', c_4', c_5')$ and $t_{*,s}$ as (u_1, u_2).
  2. Decrypt u_2 to obtain k_2 and decrypt u_1 to recover $(x_1, g_1^z, g_1^{x_2 \cdot z})$.
  3. Output 1 if the following equalities hold, and output 0 otherwise.

$$\hat{e}(g_1^z, \frac{c_4'}{\frac{c_2'}{\mathsf{H}_1(\hat{e}(c_1', \mathsf{H}_2(w)^{x_1})) \cdot w}}) = \hat{e}(g_1^{x_2 \cdot z}, c_3') \quad (3)$$

$$\hat{e}(g_1, \frac{c_2'}{\mathsf{H}_1(\hat{e}(c_1', \mathsf{H}_2(w)^{x_1})) \cdot w}) = \hat{e}(c_1', \mathsf{H}_2(w)) \quad (4)$$

$$c_5' = \mathsf{H}_3(\alpha||\mathsf{H}_2(w)||\hat{e}(c_1', \mathsf{H}_2(w)^{x_1})), \quad (5)$$

  where $\alpha = c_1'||c_2'||c_3'||c_4'$.

REMARK 3. *The element c_5 in the ciphertext is solely for the purpose of achieving IND-CCA security against an outside attacker. To achieve IND-CPA security against an outside attacker and preserve all other security properties, we can eliminate c_5 in* Encrypt*, the verification of c_5 in* Decrypt*, the verifications of c_5' in* Test_1 *and* Test_2 *in the above scheme. Moreover, the element v_5 in* $\mathsf{TrapGen}_1$ *can also be eliminated. Due to the limit of space, we put the proof of this simplified scheme in the full version of this paper.*

THEOREM 1. *The proposed ASE†† scheme is sound according to Definition 1.*

Proof. To prove the theorem, we need to show that the requirements in Definition 1 are satisfied.

Referring to the Encrypt algorithm, given a ciphertext $c_w = (c_1, c_2, c_3, c_4, c_5)$, it is straightforward to verify that the following equalities hold with probability 1.

$$\frac{c_2}{\mathsf{H}_1(\hat{e}(g_1^{x_1}, \frac{c_4}{c_3^{x_2}})) \cdot \frac{c_4}{c_3^{x_2}}} = w, \;\; \hat{e}(c_1, \mathsf{H}_2(w)) = \hat{e}(g_1, \frac{c_4}{c_3^{x_2}})),$$

$$c_5 = \mathsf{H}_3(c_1||c_2||c_3||c_4||\mathsf{H}_2(w)||\hat{e}(g_1^{x_1}, \frac{c_4}{c_3^{x_2}})).$$

Referring to the definition of the Decrypt algorithm, the above equalities means that $\mathsf{Decrypt}(c_w, SK_r) = w$. As a result, the first requirement in Definition 1 is satisfied.

As to the second requirement, we prove two things.

- If $\mathsf{Decrypt}(c, SK_r) = w$ holds, we prove that the equality $\mathsf{Test}_1(c, t_{w,s}, SK_s) = 1$ holds. Let $c = (c_1', c_2', c_3', c_4', c_5')$. Referring to the definition of the Decrypt algorithm, $\mathsf{Decrypt}(c, SK_r) = w$ implies the following equalities.

$$\frac{c_2'}{\mathsf{H}_1(\hat{e}(g_1^{x_1}, \frac{c_4'}{(c_3')^{x_2}})) \cdot \frac{c_4'}{(c_3')^{x_2}}} = w \tag{6}$$

$$\hat{e}(c_1', \mathsf{H}_2(w)) = \hat{e}(g_1, \frac{c_4'}{(c_3')^{x_2}})) \tag{7}$$

$$c_5' = \mathsf{H}_3(c_1'||c_2'||c_3'||c_4'||\mathsf{H}_2(w)||\hat{e}(g_1^{x_1}, \frac{c_4'}{(c_3')^{x_2}})). \tag{8}$$

Based on these equalities and the definition of $t_{w,s}$, we have the following.

$$\begin{aligned}
\frac{\hat{e}(v_3, c_4')}{\hat{e}(v_4, c_3')} &= \frac{\hat{e}(g_1^{x_1 y}, c_4')}{\hat{e}(g_1^{x_1 y}, (c_3')^{x_2})} \\
&= \hat{e}(g_1^{x_1 y}, \frac{c_4'}{(c_3')^{x_2}}) \\
&= \hat{e}(c_1', v_0)
\end{aligned}$$

$$\begin{aligned}
& v_2 \cdot \hat{e}(v_3, \frac{c_2'}{\mathsf{H}_1(\hat{e}(c_1', v_1))}) \\
= \;& \hat{e}(g_1^{x_1 y}, w^{-1}) \cdot \hat{e}(g_1^{x_1 y}, \frac{c_2'}{\mathsf{H}_1(\hat{e}(c_1', \mathsf{H}_2(w)^{x_1}))}) \\
= \;& \hat{e}(g_1^{x_1 y}, w^{-1}) \cdot \hat{e}(g_1^{x_1 y}, \frac{c_2'}{\mathsf{H}_1(\hat{e}(g_1^{x_1}, \frac{c_4'}{(c_3')^{x_2}})) \cdot w}) \\
= \;& \hat{e}(g_1^{x_1 y}, \frac{c_4'}{(c_3')^{x_2}}) \\
= \;& \hat{e}(c_1', v_0)
\end{aligned}$$

$$c_5' = \mathsf{H}_3(c_1'||c_2'||c_3'||c_4'||v_5||\hat{e}(c_1', v_1))$$

Based on the definition of Test_1, we can conclude that $\mathsf{Test}_1(c, t_{w,s}, SK_s) = 1$ holds.

- If $\mathsf{Test}_1(c, t_{w,s}, SK_s) = 1$ holds, we prove that the equality $\mathsf{Decrypt}(c, SK_r) = w$ holds. Let $c = (c_1', c_2', c_3', c_4', c_5')$, we can re-write it in the following form.

$$c_1' = g_1^{r_1'}, \; c_2' = \mathsf{H}_1(\hat{e}(g_1^{x_1}, \mathsf{H}_2(w)^{r_1'})) \cdot \mathsf{H}_2(w)^{r_1'} \cdot w^*,$$

$$c_3' = g_2^{r_2'}, \; c_4' = g_2^{x_2 r_2'} \cdot \mathsf{H}_2(w)^{r_1'} \cdot w^\dagger, \; c_5' = c_5',$$

for some $r_1', r_2', w^*, w^\dagger$. The equalities associated with labels (1) and (2) in the definition of Test_1 lead to the following equalities.

$$\frac{\hat{e}(g_1^{x_1 y}, g_2^{x_2 r_2'} \cdot \mathsf{H}_2(w)^{r_1'} \cdot w^\dagger)}{\hat{e}(g_1^{x_1 x_2 y}, g_2^{r_2'})} = \hat{e}(g_1^{r_1'}, \mathsf{H}_2(w)^{x_1 y}),$$

$$\begin{aligned}
& \hat{e}(g_1^{x_1 y}, w^{-1}) \cdot \hat{e}(g_1^{x_1 y}, \frac{\mathsf{H}_1(\hat{e}(g_1^{x_1}, \mathsf{H}_2(w)^{r_1'})) \cdot \mathsf{H}_2(w)^{r_1'} \cdot w^*}{\mathsf{H}_1(\hat{e}(g_1^{r_1'}, \mathsf{H}_2(w)^{x_1}))}) \\
= \;& \hat{e}(g_1^{r_1'}, \mathsf{H}_2(w)^{x_1 y}),
\end{aligned}$$

$$c_5' = \mathsf{H}_3(c_1'||c_2'||c_3'||c_4'||\mathsf{H}_2(w)||\hat{e}(c_1', \mathsf{H}_2(w)^{x_1})).$$

The first equality implies $w^\dagger = 1$, and the second equality implies that $w^* = w$. Clearly, $\mathsf{Decrypt}(c, SK_r) = w$ holds.

As to the third requirement, we prove two things.

- If $\mathsf{Decrypt}(c, SK_r) = w$ holds, we prove that the equality $\mathsf{Test}_2(c, w, t_{*,s}, SK_s) = 1$ holds. Let $c = (c_1', c_2', c_3', c_4', c_5')$. Based on these equalities associated with labels (6),(7),(8)

and the definition of $t_{*,s}$, we have the following.

$$\begin{aligned}\hat{e}(g_1^z, \frac{c_4'}{\frac{c_2'}{\mathsf{H}_1(\hat{e}(c_1', \mathsf{H}_2(w)^{x_1}))\cdot w}}) &= \hat{e}(g_1^z, \frac{c_4'}{\frac{c_2'}{\mathsf{H}_1(\hat{e}(g_1^{x_1}, \frac{c_4'}{(c_3')^{x_2}}))\cdot w}}) \\ &= \hat{e}(g_1^z, \frac{c_4'}{\frac{c_4'}{(c_3')^{x_2}}}) \\ &= \hat{e}(g_1^z, (c_3')^{x_2}) \\ &= \hat{e}(g_1^{x_2\cdot z}, c_3')\end{aligned}$$

$$\begin{aligned}&\hat{e}(g_1, \frac{c_2'}{\mathsf{H}_1(\hat{e}(c_1', \mathsf{H}_2(w)^{x_1}))\cdot w}) \\ =\ &\hat{e}(g_1, \frac{c_2'}{\mathsf{H}_1(\hat{e}(g_1^{x_1}, \frac{c_4'}{(c_3')^{x_2}}))\cdot w}) \\ =\ &\hat{e}(g_1, \frac{c_4'}{(c_3')^{x_2}}) \\ =\ &\hat{e}(c_1', \mathsf{H}_2(w))\end{aligned}$$

$$c_5' = \mathsf{H}_3(c_1'||c_2'||c_3'||c_4'||\mathsf{H}_2(w)||\hat{e}(c_1', \mathsf{H}_2(w)^{x_1})).$$

As a result, the equalities associated with labels (3), (4), and (5) in the definition of Test_2 hold, we can conclude that $\mathsf{Test}_2(c, w, t_{*,s}, SK_s) = 1$.

- If $\mathsf{Test}_2(c, w, t_{*,s}, SK_s) = 1$, we prove that $\mathsf{Decrypt}(c, SK_r) = w$. Let $c = (c_1', c_2', c_3', c_4', c_5')$, we can re-write it in the following form.

$$c_1' = g_1^{r_1'},\ c_2' = \mathsf{H}_1(\hat{e}(g_1^{x_1}, \mathsf{H}_2(w)^{r_1'}))\cdot \mathsf{H}_2(w)^{r_1'}\cdot w^*,$$

$$c_3' = g_2^{r_2'},\ c_4' = g_2^{x_2 r_2'}\cdot \mathsf{H}_2(w)^{r_1'}\cdot w^\dagger,\ c_5' = c_5',$$

for some $r_1', r_2', w^*, w^\dagger$. The equalities associated with labels (3), (4), and (5) in the definition of Test_2 lead to the following equalities.

$$\hat{e}(g_1^z, \frac{g_2^{x_2 r_2'}\cdot \mathsf{H}_2(w)^{r_1'}\cdot w^\dagger}{\frac{\mathsf{H}_1(\hat{e}(g_1^{x_1}, \mathsf{H}_2(w)^{r_1'}))\cdot \mathsf{H}_2(w)^{r_1'}\cdot w^*}{\mathsf{H}_1(\hat{e}(g_1^{r_1'}, \mathsf{H}_2(w)^{x_1}))\cdot w}}) = \hat{e}(g_1^{x_2\cdot z}, g_2^{r_2'}),$$

$$\hat{e}(g_1, \frac{\mathsf{H}_1(\hat{e}(g_1^{x_1}, \mathsf{H}_2(w)^{r_1'}))\cdot \mathsf{H}_2(w)^{r_1'}\cdot w^*}{\mathsf{H}_1(\hat{e}(g_1^{r_1'}, \mathsf{H}_2(w)^{x_1}))\cdot w}) = \hat{e}(g_1^{r_1'}, \mathsf{H}_2(w)),$$

$$c_5' = \mathsf{H}_3(c_1'||c_2'||c_3'||c_4'||\mathsf{H}_2(w)||\hat{e}(c_1', \mathsf{H}_2(w)^{x_1})).$$

The second equality implies that $w^* = w$. Based on this fact, it is straightforward to verify that the first equality implies $w^\dagger = 1$. Based on the proof of the first requirement, it is clear that $\mathsf{Decrypt}(c, SK_r) = w$ holds.

All three conditions required by Definition 1 hold, the theorem follows. □

4.3 Security Analysis

Let the pairing parameters $param = (\mathbb{G}, \mathbb{G}_T, \hat{e}, p)$ be defined in the same way as in the $\mathsf{rKeyGen}$ algorithm. We briefly review the Decision Linear (DLIN) assumption [3] and Bilinear Diffie-Hellman (BDH) assumption [5].

The DLIN assumption is as follows: any P.P.T. attacker adv can only distinguish T_0 and T_1 with a negligible advantage $|\Pr[Adv(param, T_0) = 1] - \Pr[Adv(param, T_1) = 1]|$, where $x, y \in_R \mathbb{Z}_p$, $g_a, g_b, g_c, \theta \in_R \mathbb{G}$, and

$$T_0 = (g_a,\ g_b,\ g_c,\ g_a^x,\ g_b^y,\ g_c^{x+y}),$$

$$T_1 = (g_a,\ g_b,\ g_c,\ g_a^x,\ g_b^y,\ \theta).$$

The BDH assumption is as follows: given $(param, g, g^x, g^y, g^z)$ where $x, y, z \in_R \mathbb{Z}_p$ and g is a generator of $\mathbb{G}$, any P.P.T. attacker adv can only compute $\hat{e}(g, g)^{x\cdot y\cdot z}$ with a negligible probability.

In proving one of the theorems, we also use the assumption that inverting the bilinear map $\hat{e}$ is hard, and this problem has been shown to be equivalent to the discrete logarithm problem in $\mathbb{G}$ and $\mathbb{G}_T$ [10, 15]. Certainly, this assumption is weaker than both DLIN and BDH assumptions.

THEOREM 2. *The proposed scheme achieves CI security against a curious Type-I server $\mathcal{S}_1$ in the random oracle model given: (1) $(\mathsf{KeyGen}', \mathsf{Encrypt}', \mathsf{Decrypt}')$ and $(\mathsf{Enc}, \mathsf{Dec})$ are IND-CPA secure; (2) the DLIN and BDH assumptions.*

Proof. Under Definition 3, let Game0 be the attack game defined in Figure 2. Let the attacker's advantage be ϵ_0.

Next, consider a game Game1, where the challenger performs in the same way as in Game0, except for the following. For any $\mathsf{TrapGen}_2$ query with the input PK_s, the challenger returns $t_{*,s} = (u_1, u_2)$, generated as follows: select $k_2', k_2'' \in_R \mathcal{M}$, $\gamma \in_R \mathbb{Z}_p$ and $\alpha, \beta \in_R \mathbb{G}$, compute $u_1 = \mathsf{Enc}(\gamma||\alpha||\beta, k_1')$ and $u_2 = \mathsf{Encrypt}'(k_2'', PK_s)$. In this game, let the attacker's advantage be ϵ_1. Based on the IND-CPA definition for encryption schemes, $|\epsilon_1 - \epsilon_0|$ is negligible if both $(\mathsf{KeyGen}', \mathsf{Encrypt}', \mathsf{Decrypt}')$ and $(\mathsf{Enc}, \mathsf{Dec})$ are IND-CPA secure.

Next, consider a game Game2, where the challenger performs in the same way as in Game1, except for the following.

- At the beginning of the game, the challenger chooses $y^* \in_R \mathbb{Z}_p$ and constructs a list for the random oracle H_2. If H_2 is queried with input w, the challenger first checks the list. If there is already a hash value for w, the challenger returns this value; otherwise, the challenger chooses $r_w \in_R \mathbb{Z}_p$ and returns $g_1^{r_w}$ as the hash value, and then adds $(w, g_1^{r_w}, r_w)$ to the list.
- For a $\mathsf{TrapGen}_1$ query with input w and PK_s, the challenger generates the message-dependent trapdoor $t_{w,s} = (v_7, v_8)$ as follows.

$$y_w \in_R \mathbb{Z}_p,\ v_0 = g_1^{x_1\cdot y^*\cdot r_w\cdot y_w},\ v_1 = g_1^{x_1\cdot r_w},$$

$$v_2 = \hat{e}(g_1^{x_1\cdot y^*\cdot y_w}, w^{-1}), v_3 = g_1^{x_1\cdot y^*\cdot y_w}, v_4 = g_1^{x_1\cdot x_2\cdot y^*\cdot y_w},$$

$$v_5 = g_1^{r_w},\ k_1 \in_R \mathcal{M},\ v_8 = \mathsf{Encrypt}'(k_1, PK_s),$$

$$v_7 = \mathsf{Enc}(v_0||v_1||v_2||v_3||v_4||v_5, k_1).$$

In this game, let the attacker's advantage be ϵ_2. Given that H_2 is modeled as a random oracle, this game is identical to Game1 so that $\epsilon_2 = \epsilon_1$.

Next, consider a game Game3, where the challenger performs in the same way as in Game2, except that the challenge $c_{w_b} = (c_1, c_2, c_3, c_4, c_5)$ is generated as follows.

$$r_1, r_2 \in_R \mathbb{Z}_p,\ R \in_R \mathbb{G}_2,\ R' \in_R \{0,1\}^\lambda,\ c_1 = g_1^{r_1},$$

$$c_2 = R \cdot \mathsf{H}_2(w_b)^{r_1} \cdot w_b, c_3 = g_2^{r_2}, c_4 = g_2^{x_2 \cdot r_2} \cdot \mathsf{H}_2(w_b)^{r_1}, c_5 = R'.$$

In Game3, let the attacker's advantage be ϵ_3. If H_1 and H_3 are modeled as random oracles, then Game3 differs from Game2 only if $\hat{e}(g_1^{x_1}, \mathsf{H}_2(w_b)^{r_1})$ has been queried to H_1 or $*||\hat{e}(g_1^{x_1}, \mathsf{H}_2(w)^{r_1})||*$ has been queried to H_3 where the * can be anything. Based on the Difference lemma [11] and CLAIM 1 (stated and proven below), $|\epsilon_3 - \epsilon_2|$ is negligible based on the BDH assumption.

Next, consider a new game Game4, where the challenger performs in the same way as in Game3, except that the challenge $c_{w_b} = (c_1, c_2, c_3, c_4, c_5)$ is generated as follows.

$$r_1, r_2 \in_R \mathbb{Z}_p,\ R \in_R \mathbb{G}_2,\ R' \in_R \{0,1\}^\lambda,\ c_1 = g_1^{r_1},\ c_2 = R,$$

$$c_3 = g_2^{r_2},\ c_4 = g_2^{x_2 r_2} \cdot \mathsf{H}_2(w_b)^{r_1},\ c_5 = R'.$$

Clearly, Game4 is identical to Game3. Let the attacker's advantage be ϵ_4, so that we have $\epsilon_4 = \epsilon_3$. Based on its definition, ϵ_4 can also be regarded as the attacker's advantage in distinguishing X_0 and X_1, where $x_1, x_2, y^*, r_1, r_2 \in_R \mathbb{Z}_p$. Other public parameters (e.g. pairing parameters and w_0, w_1) are described in the game.

- $X_0 = (g_1, g_2, g_1^{x_1 \cdot y^*}, g_1^{x_1}, g_2^{x_2}, g_1^{x_1 \cdot x_2 \cdot y^*}, g_1^{r_1}, g_2^{r_2}, g_2^{x_2 r_2} \cdot \mathsf{H}_2(w_0)^{r_1})$
- $X_1 = (g_1, g_2, g_1^{x_1 \cdot y^*}, g_1^{x_1}, g_2^{x_2}, g_1^{x_1 \cdot x_2 \cdot y^*}, g_1^{r_1}, g_2^{r_2}, g_2^{x_2 r_2} \cdot \mathsf{H}_2(w_1)^{r_1})$

Let this distinguishing problem be referred to as D1.

Reduction step 2a. D1 is equivalent to distinguish Y_0 and Y_1, where $x_2, r_1, r_2 \in_R \mathbb{Z}_p$ and $h_1 \in_R \mathbb{G}$. Note that $g_1^{x_1 \cdot y^*}$ is set to be h_1 and $g_1^{x_1}$ is removed.

- $Y_0 = (g_1,\ g_2,\ h_1,\ g_2^{x_2},\ h_1^{x_2},\ g_1^{r_1}, g_2^{r_2},\ g_2^{x_2 r_2} \cdot \mathsf{H}_2(w_0)^{r_1})$
- $Y_1 = (g_1,\ g_2,\ h_1,\ g_2^{x_2},\ h_1^{x_2},\ g_1^{r_1}, g_2^{r_2},\ g_2^{x_2 r_2} \cdot \mathsf{H}_2(w_1)^{r_1})$

Let this distinguishing problem be referred to as D2.

Reduction step 2b. D2 can be reduced to distinguish Z_0 and Z_1, where $x_2, r_1, r_2 \in_R \mathbb{Z}_p$, $h_1, h_2, \delta \in_R \mathbb{G}$.

- $Z_0 = (g_1,\ g_2,\ h_1,\ h_2,\ g_2^{x_2},\ h_1^{x_2},\ g_1^{r_1},\ g_2^{r_2},\ g_2^{x_2 r_2} \cdot h_2^{r_1})$
- $Z_1 = (g_1,\ g_2,\ h_1,\ h_2,\ g_2^{x_2},\ h_1^{x_2},\ g_1^{r_1},\ g_2^{r_2},\ \delta)$

Let this distinguishing problem be referred to as D3.

Reduction step 2c. D3 is equivalent to distinguish U_0 and U_1, where $x_2, x_3, r_1, r_2 \in_R \mathbb{Z}_p$, $h_1, \delta \in_R \mathbb{G}$. Note that h_2 is replaced by $g_2^{x_3}$.

- $U_0 = (g_1,\ g_2,\ h_1,\ g_2^{x_3},\ g_2^{x_2},\ h_1^{x_2},\ g_1^{r_1},\ g_2^{r_2},\ g_2^{x_2 r_2} \cdot g_2^{x_3 r_1})$
- $U_1 = (g_1,\ g_2,\ h_1,\ g_2^{x_3},\ g_2^{x_2},\ h_1^{x_2},\ g_1^{r_1},\ g_2^{r_2},\ \delta)$

Let this distinguishing problem be referred to as D4.

Reduction step 2d. D4 is equivalent to distinguish V_0 and V_1, where $\alpha, x_3, r_1, r_2 \in_R \mathbb{Z}_p$, $h_1, h_2, \delta \in_R \mathbb{G}$. Note that x_2 is set to be $\alpha + x_3$.

- $V_0 = (g_1, g_2, h_1, g_2^{x_3}, g_2^{\alpha+x_3}, h_1^{\alpha+x_3}, g_1^{r_1}, g_2^{r_2}, g_2^{\alpha r_2 + x_3(r_1+r_2)})$
- $V_1 = (g_1,\ g_2,\ h_1,\ g_2^{x_3},\ g_2^{\alpha+x_3},\ h_1^{\alpha+x_3},\ g_1^{r_1},\ g_2^{r_2},\ \delta)$

Let this distinguishing problem be referred to as D5.

Reduction step 2e. D5 can be reduced to distinguish W_0 and W_1, where $x_3, r_1, r_2 \in_R \mathbb{Z}_p$, $h_1, h_2, \delta \in_R \mathbb{G}$. The reduction is based on the fact that it is straightforward to construct a D5's instance from W_0 and W_1.

- $W_0 = (g_1,\ g_2,\ h_1,\ g_2^{x_3},\ h_1^{x_3},\ g_1^{r_1},\ g_2^{r_2},\ g_2^{x_3(r_1+r_2)})$
- $W_1 = (g_1,\ g_2,\ h_1,\ g_2^{x_3},\ h_1^{x_3},\ g_1^{r_1},\ g_2^{r_2},\ \delta)$

Let this distinguishing problem be referred to as D6.

Reduction step 2f. D6 can be reduced to the DLIN problem in $\mathbb{G}$. The reduction is because from a DLIN instance we can construct a D6 instance as follows, where $\gamma \in_R \mathbb{Z}_p$.

- $T_0' = (g_a,\ g_b,\ g_b^{\gamma},\ g_c,\ g_c^{\gamma},\ g_a^{x},\ g_b^{y},\ g_c^{x+y})$
- $T_1' = (g_a,\ g_b,\ g_b^{\gamma},\ g_c,\ g_c^{\gamma},\ g_a^{x},\ g_b^{y},\ \theta)$

Based on all above reductions, in Game4, the attacker's advantage ϵ_4 is negligible based on the DLIN assumption. As a result, ϵ_0 is negligible based on all the assumptions mentioned in the theorem. □

CLAIM 1. *In Game3 of the proof of Theorem 2, the attacker can only succeed in querying* H_1 *with* $\hat{e}(g_1^{x_1}, \mathsf{H}_2(w_b)^{r_1})$ *for a negligible probability based on the BDH assumption.*

Proof. In Game3, besides the public parameters, the challenger only needs $(g_1^{x_1 \cdot y^*}, g_1^{x_1 \cdot x_2 \cdot y^*})$ in order to answer the attacker's oracle queries. The computational problem (referred to as P1) is defined as follows.

- Input: $(g_1, g_2, g_1^{x_1 \cdot y^*}, \mathsf{H}_2(w_b), g_1^{x_1}, g_2^{x_2}, g_1^{x_1 \cdot x_2 \cdot y^*}, g_1^{r_1}, g_2^{r_2}, g_2^{x_2 r_2} \cdot \mathsf{H}_2(w_b)^{r_1})$, where $x_1, x_2, y^*, r_1, r_2 \in_R \mathbb{Z}_p$. Other public parameters are described in the game.
- Output: $\hat{e}(g_1^{x_1}, \mathsf{H}_2(w_b)^{r_1})$.

The problem P1 can be equivalently rephrased as follows, referred to as P2. Note that $g_1^{x_1 \cdot y^*}$ is set to be h_1 and $\mathsf{H}_2(w_b)$ is set to be h_2.

- Input: $(g_1, g_2, h_1, h_2, g_1^{x_1}, g_2^{x_2}, h_1^{x_2}, g_1^{r_1}, g_2^{r_2}, g_2^{x_2 r_2} \cdot h_2^{r_1})$, where $x_1, x_2, r_1, r_2 \in_R \mathbb{Z}_p$, $h_1 \in_R \mathbb{G}$ and $h_2 \in_R \mathbb{G}$.
- Output: $\hat{e}(g_1^{x_1}, h_2^{r_1})$ or $\hat{e}(g_1^{x_1}, g_2^{x_2 r_2})$.

The problem P2 can be reduced to the following problem (referred to as P3).

- Input: $(g_1, g_2, g_1^{x_1}, g_2^{x_2}, g_1^{x_2}, g_2^{r_2})$, where $x_1, x_2, r_1, r_2 \in_R \mathbb{Z}_p$.

- Output: $\hat{e}(g_1^{x_1}, g_2^{x_2 r_2})$.

The reduction is based on the fact that, from a P3's instance $(g_1,\ g_2,\ g_1^{x_1},\ g_2^{x_2},\ g_1^{x_2}\ g_2^{r_2})$, we can construct a P2's instance $(g_1, g_2, g_1^{\alpha}, g_2^{\beta - r_2}, g_1^{x_1}, g_2^{x_2}, (g_1^{x_2})^{\alpha}, g_1^{x_2+\gamma}, g_2^{r_2}, g_2^{x_2\beta+\beta\gamma-r_2\gamma})$ for $\alpha, \beta, \gamma \in_R \mathbb{Z}_p$. Furthermore, the problem P3 can be reduced to the following problem (referred to as P4).

- Input: $(g_2,\ g_2^{x_1},\ g_2^{x_2},\ g_2^{r_2})$, where $x_1, x_2, r_2 \in_R \mathbb{Z}_p$.
- Output: $\hat{e}(g_2, g_2)^{x_1 \cdot x_2 \cdot r_2}$.

The reduction is based on the fact that, from a P4's instance $(g_2,\ g_2^{x_1},\ g_2^{x_2},\ g_2^{r_2})$, we can construct a P2's instance $(g_2^{\tau},\ g_2,\ g_2^{x_1 \cdot \tau},\ g_2^{x_2},\ g_2^{x_2 \cdot \tau}\ g_2^{r_2})$ for $\alpha, \beta, \gamma \in_R \mathbb{Z}_p$. P4 is indeed the BDH problem. The claim now follows. □

THEOREM 3. *The proposed scheme achieves IND-CCA security against an outside attacker in the random oracle model given: (1)* $(\mathsf{KeyGen}', \mathsf{Encrypt}', \mathsf{Decrypt}')$ *and* $(\mathsf{Enc}, \mathsf{Dec})$ *are IND-CPA secure; (2) the DLIN and BDH assumptions.*

Proof. Under Definition 2, let Game0 be the attack game defined in Figure 1. Let the attacker's advantage be ϵ_0.

Next, consider a game Game1, where the challenger performs in the same way as in Game0, except for the following. For any $\mathsf{TrapGen}_1$ query with the input w and PK_s, the challenger returns $t_{w,s} = (v_7, v_8)$, generated as follows: select $v_0, v_1, v_2, v_3, v_4, v_5, k_1$ uniformly at random from their corresponding domains, compute

$$v_7 = \mathsf{Enc}(v_0||v_1||v_2||v_3||v_4||v_5, k_1)$$

and $v_8 = \mathsf{Encrypt}'(k_1, PK_s)$. In this game, let the attacker's advantage be ϵ_1. Based on the IND-CPA definition for encryption schemes, $|\epsilon_1 - \epsilon_0|$ is negligible if both $(\mathsf{KeyGen}', \mathsf{Encrypt}', \mathsf{Decrypt}')$ and $(\mathsf{Enc}, \mathsf{Dec})$ are IND-CPA secure.

Next, consider a game Game2, where the challenger performs in the same way as in Game1, except for the following. For any $\mathsf{TrapGen}_2$ query with the input PK_s, the challenger returns $t_{*,s} = (u_1, u_2)$ which is generated as follows: select $k_2', k_2'' \in_R \mathcal{M}$, $\gamma \in_R \mathbb{Z}_p$ and $\alpha, \beta \in_R \mathbb{G}_1$, compute $u_1 = \mathsf{Enc}(\gamma||\alpha||\beta, k_2')$ and $u_2 = \mathsf{Encrypt}'(k_2'', PK_s)$. In this game, let the attacker's advantage be ϵ_2. Based on the IND-CPA definition for encryption schemes, $|\epsilon_2 - \epsilon_1|$ is negligible if both $(\mathsf{KeyGen}', \mathsf{Encrypt}', \mathsf{Decrypt}')$ and $(\mathsf{Enc}, \mathsf{Dec})$ are IND-CPA secure.

Next, consider a game Game3, where the challenger performs in the same way as in Game2, except that it answers the $\mathsf{Decrypt}$ oracle as follows. Given a ciphertext c, the challenger parses it as $(c_1, c_2, c_3, c_4, c_5)$ and performs as follows.

1. Compute $w = \frac{c_2}{\mathsf{H}_1(\hat{e}(g_1^{x_1}, \frac{c_4}{c_3^{x_2}})) \cdot \frac{c_4}{c_3^{x_2}}}$.
2. If $c_1||c_2||c_3||c_4||\mathsf{H}_2(w)||\hat{e}(g_1^{x_1}, \frac{c_4}{c_3^{x_2}})$ has not been queried to H_3 or w has not been queried to H_2, abort by outputting an error symbol $\perp$. In addition, if there are different inputs to either oracle and result in the same output, abort by outputting an error symbol $\perp$.
3. If $c_5 = \mathsf{H}_3(c_1||c_2||c_3||c_4||\mathsf{H}_2(w)||\hat{e}(g_1^{x_1}, \frac{c_4}{c_3^{x_2}}))$ and $\hat{e}(c_1, \mathsf{H}_2(w)) = \hat{e}(g_1, \frac{c_4}{c_3^{x_2}}))$ output w. Otherwise, abort by outputting an error symbol $\perp$.

This game is identical to Game2 unless the following event occurs: there is a ciphertext c, the $\mathsf{Decrypt}$ algorithm outputs w but the challenger returns $\perp$. If H_2 and H_3 are modeled as random oracles, the event occurs with a negligible probability. In this game, let the attacker's advantage be ϵ_3. Based on the Difference lemma [11], $|\epsilon_3 - \epsilon_2|$ is negligible in the random oracle model.

Next, consider a game Game4, where the challenger performs in the same way as in Game3, except that it answers the $\mathsf{Decrypt}$ oracle as follows. Given a ciphertext c, the challenger parses it as $(c_1, c_2, c_3, c_4, c_5)$ and performs as follows.

1. Check whether there is an input $c_1||c_2||c_3||c_4||\alpha_1||\alpha_2$ to H_3, where $\alpha_1 \in \mathbb{G}$ and $\alpha_2 \in \mathbb{G}_T$ and there is an input $w' \in \mathbb{G}$ to H_2 such that
$$c_5 = \mathsf{H}_3(c_1||c_2||c_3||c_4||\alpha_1||\alpha_2), \mathsf{H}_2(w') = \alpha_1.$$
If there is no such inputs or there are more than one inputs satisfying the equalities, aborts by outputting an error symbol $\perp$.
2. Compute $\beta = \frac{c_2}{\mathsf{H}_1(\alpha_2) \cdot w'}$.
3. Check whether the following equalities hold.
$$\hat{e}(g_1^{x_1}, \beta) = \alpha_2,\ \hat{e}(\frac{c_4}{\beta}, g_2) = \hat{e}(g_2^{x_2}, c_3),$$
$$\hat{e}(c_1, \mathsf{H}_2(w')) = \hat{e}(g_1, \beta).$$
If so, output w', otherwise aborts by outputting an error symbol $\perp$.

This game is identical to Game3, but the receiver's private keys are not required to answer the oracle queries. In this game, let the attacker's advantage be ϵ_4, so that $\epsilon_3 = \epsilon_4$.

Next, consider a game Game5, where the challenger performs in the same way as in Game4, except that the challenge $c_{w_b} = (c_1, c_2, c_3, c_4, c_5)$ is generated as follows.

$$r_1, r_2 \in_R \mathbb{Z}_p,\ R \in_R \mathbb{G}_2,\ R' \in_R \{0,1\}^{\lambda},\ c_1 = g_1^{r_1},$$

$$c_2 = R \cdot \mathsf{H}_2(w_b)^{r_1} \cdot w_b, c_3 = g_2^{r_2}, c_4 = g_2^{x_2 \cdot r_2} \cdot \mathsf{H}_2(w_b)^{r_1}, c_5 = R'.$$

In this game, let the attacker's advantage be ϵ_5. Note the fact that the challenger answers $\mathsf{Decrypt}$ oracle access without the knowledge of the receiver's private keys (i.e. $\mathsf{Decrypt}$ oracle access gives the attacker no actual privilege), therefore an outside attacker infact has less privilege than a curious Type-I server. Based on the analysis from Game3 and the following games in the proof of Theorem 2, we can conclude that ϵ_5 is negligible based on the the DLIN assumption and the BDH assumption. The theorem now follows. □

THEOREM 4. *The proposed scheme achieves CI security against a hybrid server* $\mathcal{S}_h$ *in the random oracle model given: (1)* $(\mathsf{KeyGen}', \mathsf{Encrypt}', \mathsf{Decrypt}')$ *and* $(\mathsf{Enc}, \mathsf{Dec})$ *are IND-CPA secure; (2) the DLIN and BDH assumptions.*

Proof. Under Definition 4, suppose the attacker's advantage is ϵ_0 in the attack game defined in Figure 3. With respect to the proposed scheme, it is clear that, given a master trapdoor $t_{*,s}$, an attacker can simulate the $\mathsf{TrapGen}_1$ oracle by itself. As a result, ϵ_0 can also be regarded as the attacker's advantage in distinguishing X_0 and X_1, where $x_1, x_2, r_1, r_2, r_1', r_2', z \in_R \mathbb{Z}_p$, and $w_0, w_1 \in_R \mathbb{G}$, and any $c_1, c_2, c_3, c_4, c_1', c_2', c_3', c_4'$. Other public parameters are described in the game.

- $X_0 = (x_1, g_1,\ g_2,\ g_1^{x_1},\ g_2^{x_2},\ g_1^z,\ g_1^{x_2 \cdot z},\ g_1^{r_1},$
  $\mathsf{H}_1(\hat{e}(g_1^{x_1}, \mathsf{H}_2(w_0)^{r_1})) \cdot \mathsf{H}_2(w_0)^{r_1} \cdot w_0, g_2^{r_2}, g_2^{x_2 r_2} \cdot \mathsf{H}_2(w_0)^{r_1},$
  $\mathsf{H}_3(c_1||c_2||c_3||c_4||\mathsf{H}_2(w_0)||\hat{e}(g_1^{x_1}, \mathsf{H}_2(w_0)^{r_1})),\ g_1^{r_1'},$
  $\mathsf{H}_1(\hat{e}(g_1^{x_1}, \mathsf{H}_2(w_0)^{r_1'})) \cdot \mathsf{H}_2(w_0)^{r_1'} \cdot w_0, g_2^{r_2'}, g_2^{x_2 r_2'} \cdot \mathsf{H}_2(w_0)^{r_1'},$
  $\mathsf{H}_3(c_1'||c_2'||c_3'||c_4'||\mathsf{H}_2(w_0)||\hat{e}(g_1^{x_1}, \mathsf{H}_2(w_0)^{r_1'})))$

- $X_1 = (x_1, g_1,\ g_2,\ g_1^{x_1},\ g_2^{x_2},\ g_1^z,\ g_1^{x_2 \cdot z},\ g_1^{r_1},$
  $\mathsf{H}_1(\hat{e}(g_1^{x_1}, \mathsf{H}_2(w_0)^{r_1})) \cdot \mathsf{H}_2(w_0)^{r_1} \cdot w_0, g_2^{r_2}, g_2^{x_2 r_2} \cdot \mathsf{H}_2(w_0)^{r_1},$
  $\mathsf{H}_3(c_1||c_2||c_3||c_4||\mathsf{H}_2(w_0)||\hat{e}(g_1^{x_1}, \mathsf{H}_2(w_0)^{r_1})),\ g_1^{r_1'},$
  $\mathsf{H}_1(\hat{e}(g_1^{x_1}, \mathsf{H}_2(w_1)^{r_1'})) \cdot \mathsf{H}_2(w_1)^{r_1'} \cdot w_1, g_2^{r_2'}, g_2^{x_2 r_2'} \cdot \mathsf{H}_2(w_1)^{r_1'},$
  $\mathsf{H}_3(c_1'||c_2'||c_3'||c_4'||\mathsf{H}_2(w_1)||\hat{e}(g_1^{x_1}, \mathsf{H}_2(w_1)^{r_1'})))$

Let the above distinguishing problem be denoted as I1.

Reduction step 4a. I1 is equivalent to distinguish Y_0 and Y_1, where $x_1, x_2, r_1, r_2, r_1', r_2', z \in_R \mathbb{Z}_p$ and $w_0, w_1, h_0, h_1 \in_R \mathbb{G}$, and any $c_1, c_2, c_3, c_4, c_1', c_2', c_3', c_4'$. Note that $\mathsf{H}_2(w_0)$ is set to be h_0 and $\mathsf{H}_2(w_1)$ is set to be h_1.

- $Y_0 = (x_1,\ g_1,\ g_2,\ g_1^{x_1},\ g_2^{x_2},\ g_1^z,\ g_1^{x_2 \cdot z},\ g_1^{r_1},$
  $\mathsf{H}_1(\hat{e}(g_1^{x_1}, h_0^{r_1})) \cdot h_0^{r_1} \cdot w_0,\ g_2^{r_2},\ g_2^{x_2 r_2} \cdot h_0^{r_1},$
  $\mathsf{H}_3(c_1||c_2||c_3||c_4||h_0||\hat{e}(g_1^{x_1}, h_0^{r_1})),\ g_1^{r_1'},$
  $\mathsf{H}_1(\hat{e}(g_1^{x_1}, h_0^{r_1'})) \cdot h_0^{r_1'} \cdot w_0,\ g_2^{r_2'},\ g_2^{x_2 r_2'} \cdot h_0^{r_1'},$
  $\mathsf{H}_3(c_1'||c_2'||c_3'||c_4'||h_0||\hat{e}(g_1^{x_1}, h_0^{r_1'})))$

- $Y_1 = (x_1,\ g_1,\ g_2,\ g_1^{x_1},\ g_2^{x_2},\ g_1^z,\ g_1^{x_2 \cdot z},\ g_1^{r_1},$
  $\mathsf{H}_1(\hat{e}(g_1^{x_1}, h_0^{r_1})) \cdot h_0^{r_1} \cdot w_0,\ g_2^{r_2},\ g_2^{x_2 r_2} \cdot h_0^{r_1},$
  $\mathsf{H}_3(c_1||c_2||c_3||c_4||h_0||\hat{e}(g_1^{x_1}, h_0^{r_1})),\ g_1^{r_1'},$
  $\mathsf{H}_1(\hat{e}(g_1^{x_1}, h_1^{r_1'})) \cdot h_1^{r_1'} \cdot w_1,\ g_2^{r_2'},\ g_2^{x_2 r_2'} \cdot h_1^{r_1'},$
  $\mathsf{H}_3(c_1'||c_2'||c_3'||c_4'||h_1||\hat{e}(g_1^{x_1}, h_1^{r_1'})))$

Let this distinguishing problem be denoted as I2.

Reduction step 4b. Let Y_0' and Y_1' be the following. Straightforwardly, CLAIM 2 (stated and proven below) also implies that, for any P.P.T. attacker, given Y_1', it can only compute $\hat{e}(g_1^{x_1}, h_0^{r_1})$ and $\hat{e}(g_1^{x_1}, h_1^{r_1'})$ with a negligible probability based on the BDH assumption.

- $Y_0' = (x_1,\ g_1,\ g_2,\ g_1^{x_1},\ g_2^{x_2},\ g_1^z,\ g_1^{x_2 \cdot z},\ g_1^{r_1},\ g_2^{r_2},\ g_2^{x_2 r_2} \cdot h_0^{r_1},\ g_1^{r_1'},\ g_2^{r_2'},\ g_2^{x_2 r_2'} \cdot h_0^{r_1'})$

- $Y_1' = (x_1,\ g_1,\ g_2,\ g_1^{x_1},\ g_2^{x_2},\ g_1^z,\ g_1^{x_2 \cdot z},\ g_1^{r_1},\ g_2^{r_2},\ g_2^{x_2 r_2} \cdot h_0^{r_1},\ g_1^{r_1'},\ g_2^{r_2'},\ g_2^{x_2 r_2'} \cdot h_1^{r_1'})$

As a result, I2 is equivalent to distinguish Y_0' and Y_1' based on the BDH assumption in the random oracle model. Let this distinguishing problem be denoted as I3.

Reduction step 4c. I3 is equivalent to distinguish Z_0 and Z_1, where $x_2, r_1, r_2, r_1', r_2', z \in_R \mathbb{Z}_p$ and $h_0, h_1 \in_R \mathbb{G}$. Note that x_1, g^{x_1} are removed.

- $Z_0 = (g_1, g_2, g_2^{x_2}, g_1^z, g_1^{x_2 \cdot z}, g_1^{r_1}, g_2^{r_2}, g_2^{x_2 r_2} \cdot h_0^{r_1}, g_1^{r_1'}, g_2^{r_2'}, g_2^{x_2 r_2'} \cdot h_0^{r_1'})$

- $Z_1 = (g_1, g_2, g_2^{x_2}, g_1^z, g_1^{x_2 \cdot z}, g_1^{r_1}, g_2^{r_2}, g_2^{x_2 r_2} \cdot h_0^{r_1}, g_1^{r_1'}, g_2^{r_2'}, g_2^{x_2 r_2'} \cdot h_1^{r_1'})$

Let this distinguishing problem be denoted as I4.

Reduction step 4d. I4 can be reduced to distinguish U_0 and U_1, where $x_2, r_1', r_2', z \in_R \mathbb{Z}_p$ and $h_0, h_1 \in_R \mathbb{G}$. The reduction is based on the fact that we can construct $g_1^{r_1}, g_2^{r_2}, g_2^{x_2 r_2} \cdot h_0^{r_1}$ based on h_0.

- $U_0 = (g_1,\ g_2,\ g_2^{x_2},\ g_1^z,\ g_1^{x_2 \cdot z},\ h_0, g_1^{r_1'},\ g_2^{r_2'},\ g_2^{x_2 r_2'} \cdot h_0^{r_1'})$

- $U_1 = (g_1,\ g_2,\ g_2^{x_2},\ g_1^z,\ g_1^{x_2 \cdot z},\ h_0, g_1^{r_1'},\ g_2^{r_2'},\ g_2^{x_2 r_2'} \cdot h_1^{r_1'})$

Let this distinguishing problem be denoted as I5.

It is easy to see that I5 can be reduced to the distinguishing problem D2 defined at *Reduction step 2a.* in the proof of Theorem 2 (based on a D2's instance, by adding either $\mathsf{H}_2(w_0)$ or $\mathsf{H}_2(w_1)$, we can obtain an I5's instance). Therefore, from the proof of Theorem 2, a P.P.T. attacker can only have a negligible advantage in solving I5 based on the DLIN assumption. As a result, ϵ_0 is negligible based on all the assumptions mentioned in the theorem. □

CLAIM 2. *In the proof of Theorem 4, for any P.P.T. attacker, given either Y_0', it can only compute $\hat{e}(g_1^{x_1}, h_0^{r_1})$ and $\hat{e}(g_1^{x_1}, h_0^{r_1'})$ with a negligible probability based on the BDH assumption.*

Proof. Due to the symmetry, we only need to prove that, for any P.P.T. attacker, given either Y_0', it can only compute $\hat{e}(g_1^{x_1}, h_0^{r_1})$ with a negligible probability.

Reduction step a1. Given $(x_1, g_1, g_2, g_1^{x_1}, g_2^{x_2}, g_1^z, g_1^{x_2 \cdot z}, g_1^{r_1}, g_2^{r_2},\ g_2^{x_2 r_2} \cdot h_0^{r_1})$, we can construct Y_0^* as follows, where $\alpha, \beta \in_R \mathbb{Z}_p$. $Y_0^* = (x_1, g_1, g_2, g_1^{x_1}, g_2^{x_2}, g_1^z, g_1^{x_2 \cdot z}, g_1^{r_1}, g_2^{r_2}, g_2^{x_2 r_2} \cdot h_0^{r_1}, g_1^{r_1 \alpha}, g_2^{r_2 \alpha + \beta}, (g_2^{x_2 r_2} \cdot h_0^{r_1})^{\alpha} \cdot g_2^{x_2 \beta})$. It is easy to verify that Y_0^* has an identical distribution to that of Y_0'. So, the problem can be reduced to: given $(x_1, g_1, g_2, g_1^{x_1}, g_2^{x_2}, g_1^z, g_1^{x_2 \cdot z}, g_1^{r_1}, g_2^{r_2},\ g_2^{x_2 r_2} \cdot h_0^{r_1})$, the attacker computes $\hat{e}(g_1^{x_1}, g_2^{x_2 r_2})$.

Reduction step a2. The above problem can be reduced to: given $(x_1,\ g_1,\ g_2,\ g_1^{x_1},\ g_2^{x_2},\ g_1^z,\ g_1^{x_2 \cdot z},\ g_2^{r_2})$, the attacker computes $\hat{e}(g_1^{x_1}, g_2^{x_2 r_2})$.

Reduction step a3. The above problem can be reduced to: given $(g_1,\ g_2,\ g_2^{x_2},\ g_1^z,\ g_1^{x_2 \cdot z},\ g_2^{r_2})$, the attacker computes $\hat{e}(g_1, g_2^{x_2 r_2})$.

Reduction step a4. This above reduced problem is equivalent to: given $(f_1^{x_3}, g_2, g_2^{x_2}, f_1, f_1^{x_2}, g_2^{r_2})$, where $f_1 \in_R \mathbb{G}$ and $x_3 \in_R \mathbb{Z}_p$, the attacker can only compute $\hat{e}(f_1^{x_3}, g_2^{x_2 r_2})$ with a negligible probability. The equivalence is based on the fact that $g_1, g_1^z, g_1^{x_2 \cdot z}$ are replaced with $f_1^{x_3}, f_1, f_1^{x_2}$ respectively, which is simply a change of notation. Referring to the reduction from P3 to P4 in the proof of CLAIM 1, this can be reduced to the BDH problem. The claim follows. □

Note that this theorem implies that the proposed scheme is also secure under Definition 5.

The following theorem is straightforward, so that we skip a formal proof.

THEOREM 5. *The proposed scheme achieves universal trapdoor one-wayness under Definition 6, given: (1) the hash function H_2 is one-way; (2) the bilinear map $\hat{e}$ is one-way.*

THEOREM 6. *The proposed scheme achieves trapdoor indistinguishability property if* $(\mathsf{KeyGen}', \mathsf{Encrypt}', \mathsf{Decrypt}')$ *and* $(\mathsf{Enc}, \mathsf{Dec})$ *are IND-CPA secure.*

Proof. Under Definition 7, according to the attack game definition in Figure 5, the attacker can query $\mathsf{sKeyReq}_{t'}$ for $t' = 1$ or $t' = h$. Without loss of generality, we assume that the attacker queries $\mathsf{sKeyReq}_1$. This means $t = h$, and the challenge $t_{w_b} = (v_7, v_8)$ are defined as follows.

$$y \in_R \mathbb{Z}_p, v_0 = \mathsf{H}_2(w_b)^{x_1 y}, v_1 = \mathsf{H}_2(w_b)^{x_1}, v_2 = \hat{e}(g_1^{x_1 y}, w_b^{-1}),$$

$$v_3 = g_1^{x_1 y},\ v_4 = g_1^{x_1 x_2 y},\ v_5 = \mathsf{H}_2(w_b),\ k_1 \in_R \mathcal{M},$$

$$v_7 = \mathsf{Enc}(v_0||v_1||v_2||v_3||v_4||v_5, k_1), v_8 = \mathsf{Encrypt}'(k_1, PK_{s_h}).$$

Note the fact that revealing the receiver's private key x_1, x_2 will not affect the security of the encryption schemes, namely $(\mathsf{KeyGen}', \mathsf{Encrypt}', \mathsf{Decrypt}')$ and $(\mathsf{Enc}, \mathsf{Dec})$. Without oracle access to SK_{s_h}, the attacker can learn nothing about w_b from t_{w_b}. Then, the theorem follows. □

5. CONCLUSION

In this paper, we have formulated the concept of ASE†† and attempted to provide a comprehensive security model for the primitive. As to the relationships of various security properties, the IND-CPA security against an outside attacker is implied by the ciphertext indistinguishability property against a Type-I server and the message-dependent trapdoor indistinguishability property (to be presented in the full paper due to space limit). Compared with previous works, such as those from [7, 8, 9], our formulation defines more flexible functionalities and our security model reflects a higher level of security guarantees. The security of the proposed instantiation relies on the standard DLIN and BDH assumptions in the random oracle model, which plays an important role in the double encryption structure. It is an interesting future work to investigate an instantiation in the standard model, namely without using random oracle. With regard to trapdoor security, we have designed universal one-wayness property against all attackers. There is a possibility of replacing it with the augmented notion proposed in [2] or an even stronger notion similar to the ciphertext indistinguishability property against Type-II attacker in our security model. We consider this to be another line of future work.

6. REFERENCES

[1] J. Baek, R. Safavi-Naini, and W. Susilo. On the integration of public key data encryption and public key encryption with keyword search. In S. K. Katsikas et al., editor, *Proceedings of the 9th international conference on Information Security*, volume 4176 of *LNCS*, pages 217–232. Springer, 2006.

[2] M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In *Advances in cryptology — CRYPTO 2007*, pages 535–552. Springer, 2007.

[3] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In M. K. Franklin, editor, *Advances in Cryptology — CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, 2004.

[4] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public Key Encryption with Keyword Search. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology — EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 506–522. Springer, 2004.

[5] D. Boneh and M. K. Franklin. Identity-based encryption from the weil pairing. In J. Kilian, editor, *Advances in Cryptology — CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, 2001.

[6] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and D. Chaum, editors, *Advances in Cryptology – CRYPTO 1984*, volume 196 of *LNCS*, pages 10–18. Springer, 1985.

[7] T. Fuhr and P. Paillier. Decryptable searchable encryption. In *Proceedings of the 1st international conference on Provable security*, pages 228–236, 2007.

[8] D. Hofheinz and E. Weinreb. Correlation-resistant storage via keyword-searchable encryption. Cryptology ePrint Archive: Report 2008/423, 2008.

[9] L. Ibraimi, S. Nikova, P. H. Hartel, and W. Jonker. Public-key encryption with delegated search. In J. Lopez and G. Tsudik, editors, *Applied Cryptography and Network Security - 9th International Conference*, volume 6715 of *LNCS*, pages 532–549, 2011.

[10] K. Karabina, E. Knapp, and A. Menezes. Generalizations of verheul's theorem to asymmetric pairings. Technical Report CACR 2012-03, University of Waterloo, 2012.

[11] V. Shoup. Sequences of games: a tool for taming complexity in security proofs. http://shoup.net/papers/, 2006.

[12] D. X. Song, D. Wagner, and A. Perrig. Practical Techniques for Searches on Encrypted Data. In *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, pages 44–55, 2000.

[13] Q. Tang. Search in encrypted data: Theoretical models and practical applications. Cryptology ePrint Archive: Report 2012/648, 2012.

[14] Q. Tang, Y. Zhao, X. Chen, and H. Ma. Refine the concept of public key encryption with delegated search. Cryptology ePrint Archive: Report 2012/654, 2012.

[15] E. R. Verheul. Evidence that xtr is more secure than supersingular elliptic curve cryptosystems. *J. Cryptol.*, 17(4):277–296, 2004.

[16] R. Zhang and H. Imai. Generic combination of public key encryption with keyword search and public key encryption. In *Proceedings of the 6th international conference on Cryptology and network security*, pages 159–174. Springer, 2007.