information

MDPI

Article

Certificateless Public Key Authenticated Encryption with Keyword Search Achieving Stronger Security

Jingwei Lu [1], Hongbo Li [1,*], Jianye Huang [2], Sha Ma [1], Man Ho Allen Au [3] and Qiong Huang [1,4]

[1] College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China
[2] School of Computing and Information Technology, University of Wollongong, Wollongong, NSW 2500, Australia
[3] Department of Computing, Hong Kong Polytechnic University, Hong Kong 999077, China
[4] Guangzhou Key Laboratory of Intelligent Agriculture, Guangzhou 510642, China
* Correspondence: hongbo@scau.edu.cn

Abstract: Transforming data into ciphertexts and storing them in the cloud database is a secure way to simplify data management. *Public key encryption with keyword search* (PEKS) is an important cryptographic primitive as it provides the ability to search for the desired files among ciphertexts. As a variant of PEKS, *certificateless public key authenticated encryption with keyword search* (CLPAEKS) not only simplifies certificate management but also could resist *keyword guessing attacks* (KGA). In this paper, we analyze the security models of two recent CLPAEKS schemes and find that they ignore the threat that, upon capturing two trapdoors, the adversary could directly compare them and distinguish whether they are generated using the same keyword. To cope with this threat, we propose an improved security model and define the notion of strong trapdoor indistinguishability. We then propose a new CLPAEKS scheme and prove it to be secure under the improved security model based on the intractability of the DBDH problem and the DDH problem in the targeted bilinear group.

Keywords: encryption with keyword search; certificateless public key cryptography; keyword guessing attacks; trapdoor indistinguishability; provable security

Citation: Lu, J.; Li, H.; Huang, J.; Ma, S.; Au, M.H.A.; Huang, Q. Certificateless Public Key Authenticated Encryption with Keyword Search Achieving Stronger Security. *Information* **2023**, *14*, 142. https://doi.org/10.3390/info14030142

Academic Editor: Marco Baldi

Received: 18 January 2023
Revised: 17 February 2023
Accepted: 20 February 2023
Published: 21 February 2023

1. Introduction

Boneh et al. [1] first proposed the notion of *public key encryption with keyword search* (PEKS). As shown in Figure 1, the workflow of PEKS includes:

1. The data sender uses the file's keyword to generate the searchable ciphertext C and uploads it along with the encrypted file to the cloud server.
2. The data receiver uses its desired keyword to generate the trapdoor td and sends it to the cloud server.
3. The cloud server runs an algorithm called Test to check whether C and td contain the same keyword and returns the corresponding file to the receiver if it does. During the search, the cloud server is unable to know the keyword as well as the content of the file.

PEKS could be applied to encrypted instant messaging apps. The client-side archive of chat logs may suffer from mistaken deletion and limited storage space. Therefore, some instant messaging apps (e.g., Google Talk and Yahoo Messenger 11 Beta) support saving chat logs on a server for future retrieval. Encrypting chat logs before uploading is a proactive defense against cyber attacks and data breaches. However, encryption destroys the original features of data and thus invalidates the traditional searching methods. Downloading and decrypting all chat logs before searching seems like a solution, but this process incurs unnecessary transmission overhead. As mentioned earlier, PEKS provides an efficient way for users to search for their desired files among encrypted chat logs.

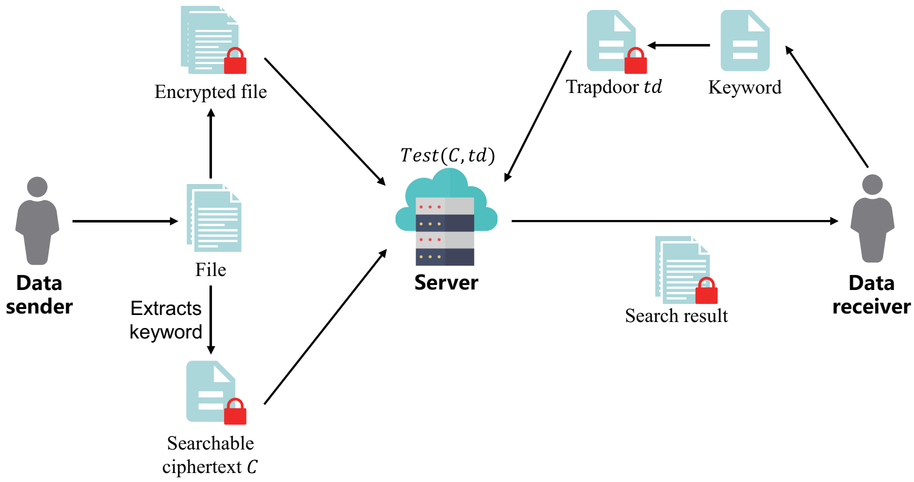

Figure 1. The general framework of PEKS

Ideally, the distribution of keywords is assumed to be uniform, and the size of keywords space is assumed to be super-polynomial. However, in practice, the distribution of keywords may be uneven, and keywords space may be much smaller. Therefore, it may be feasible for the adversary to guess the keyword of a file by launching *keyword guessing attacks* (KGA) [2,3]. As shown in Figure 2, upon capturing the trapdoor, the adversary guesses the keyword w concealed in the trapdoor td by encrypting every possible keyword and running Test algorithm. There are two types of KGA: the first type is outside KGA, launched by anyone other than the cloud server; the second type is inside KGA, launched by the cloud server. A searchable encryption scheme that could resist KGA should simultaneously satisfy ciphertext indistinguishability and trapdoor indistinguishability [4].

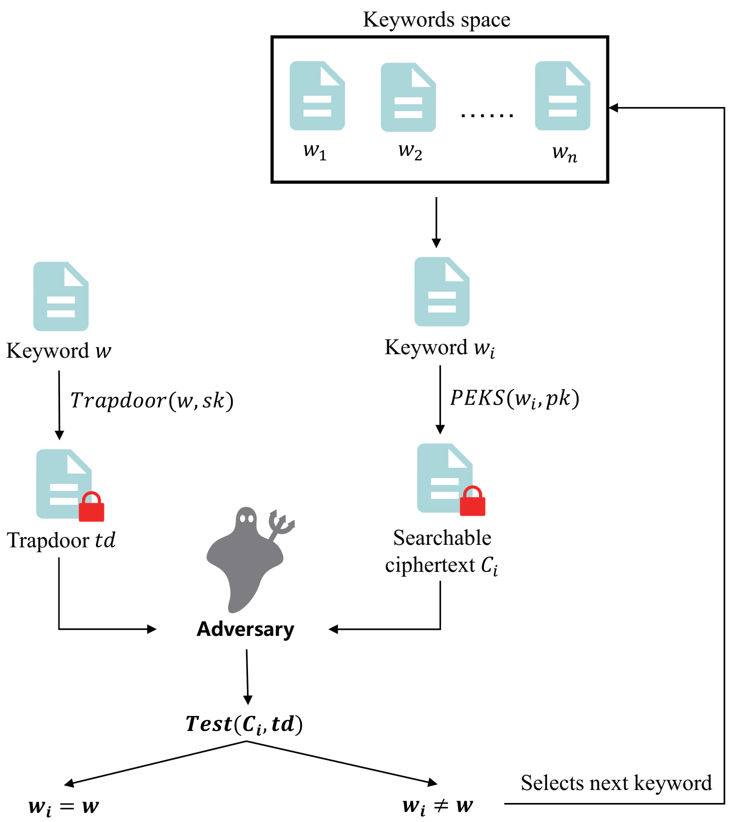

Figure 2. Keyword guessing attacks

1.1. Related Works

Song et al. [5] proposed a searchable symmetric encryption scheme. However, it suffers from problematic key distribution in symmetric key cryptography. To solve this problem, Boneh et al. [1] proposed *public key encryption with keyword search* (PEKS). However, the initial PEKS scheme [1] is vulnerable to KGA [2,3]. Rhee et al. [4] first formally defined trapdoor indistinguishability and proved that trapdoor indistinguishability is a necessary condition for a PEKS scheme to be secure against KGA. They also proposed a designated-tester PEKS (dPEKS) scheme that could resist outside KGA. Later, some improved dPEKS schemes [6,7] were proposed, but none of them could resist inside KGA.

To resist both outside and inside KGA, Wang and Tu [8] proposed a PEKS scheme based on a dual-server setting. However, their scheme is still vulnerable inside KGA if two servers collude. Huang and Li [9] proposed the first *public key authenticated encryption with keyword search* (PAEKS) scheme, which is similar to *signcrpytion* [10]. In PAEKS, the sender's secret key is involved in the ciphertext generation. As a result, the cloud server cannot launch inside KGA successfully unless it obtains either the sender's secret key or the receiver's secret key. Later, some PAEKS schemes with stronger ciphertext indistinguishability were proposed [11,12]. Pan and Li [13] proposed a PAEKS scheme with stronger trapdoor indistinguishability. However, their scheme cannot provide stronger ciphertext indistinguishability [14].

The aforementioned schemes are based on public key infrastructure and thus suffer from complicated certificate management. To solve this problem, Abdalla et al. [15] proposed the notion of *identity-based encryption with keyword search* (IBEKS), which integrates search function into *identity-based encryption* [16]. Li et al. [17] proposed the first IBEKS scheme that could resist both outside and inside KGA.

To solve the key escrow problem in IBEKS, Peng et al. [18] proposed the first searchable encryption scheme based on *certificateless public key cryptography* [19]. However, Peng et al.'s scheme are vulnerable to both outside and inside KGA. Therefore, some certificates PAEKS (CLPAEKS) schemes [20–22] were proposed. Pakniat et al. [23] analyzed the flaws of the security models defined in [20–22] and proposed an improved security model. They also presented a new CLPAEKS scheme with provable security in the proposed security model. Shiraly et al. [24] proposed an efficient CLPAEKS scheme that gets rid of the time-consuming Hash-To-Point [25] computation and bilinear pairing [16] computation.

1.2. Motivation and Contribution

We notice that in Pakniat et al.'s work [23] and Shiraly et al.'s work [24], in the games that formally define trapdoor indistinguishability, the adversary cannot query $(ID_s^\diamond, ID_r^\diamond, w_i)$ to trapdoor oracle, in which $ID_s^\diamond$ is the challenge sender, $ID_r^\diamond$ is the challenge receiver, and w_i $(i \in \{0,1\})$ is the challenge keyword.

However, in practice, the same keyword may be used for different searches. As a result, the trapdoor corresponding to $(ID_s^\diamond, ID_r^\diamond, w_i)$ may appear repeatedly. For privacy protection, it would be necessary to prevent the adversary from successfully determining whether two trapdoors are generated using the same keyword. Therefore, it is necessary to get rid of the aforementioned limitation when defining trapdoor indistinguishability.

Following are the contributions we make in this paper:

1. We propose an improved security model, in which the notion of strong trapdoor indistinguishability is defined.
2. We propose a new CLPAEKS scheme and prove it to be secure under the improved security model based on the intractability of the DBDH problem and the DDH problem in the targeted bilinear group.

2. Preliminaries

Suppose that $\mathcal{A}$ is a probabilistic-polynomial-time (PPT) adversary, $\mathbb{G}_1$ and $\mathbb{G}_T$ are cyclic groups with the same prime order p.

2.1. Bilinear Pairing

A bilinear pairing $\hat{e}:\mathbb{G}_1\times\mathbb{G}_1\rightarrow\mathbb{G}_T$ has the following features:

- Bilinearity: For any $(\varphi_1,\ \varphi_2)\in\mathbb{G}_1^2$ and any $(\eta_1,\ \eta_2)\in\mathbb{Z}_p^2$, $\hat{e}\left(\varphi_1^{\eta_1},\ \varphi_2^{\eta_2}\right)=\hat{e}(\varphi_1,\ \varphi_2)^{\eta_1\cdot\eta_2}$.
- Non-degeneracy: Suppose that φ is a generator of $\mathbb{G}_1$, $\hat{e}(\varphi,\ \varphi)\neq 1$.
- Computability: For any $(\varphi_1,\ \varphi_2)\in\mathbb{G}_1^2$, $\hat{e}(\varphi_1,\ \varphi_2)$ can be computed in polynomial time.

2.2. Decisional Diffie–Hellman (DDH) Assumption in $\mathbb{G}_T$

Given $(\varphi_t,\ \varphi_t^{\eta_1},\ \varphi_t^{\eta_2},\ Z)\in\mathbb{G}_T^4$, in which φ_t is a generator of $\mathbb{G}_T$, $(\eta_1,\ \eta_2)\in\mathbb{Z}_p^2$. $\mathcal{A}$'s aim is to determine whether $Z=\varphi_t^{\eta_1\cdot\eta_2}$ or $Z=\varphi_t^r$, in which r is randomly selected from $\mathbb{Z}_p$. The DDH assumption in $\mathbb{G}_T$ holds if $\mathcal{A}$'s advantage

$$\mathrm{Adv}_{\mathcal{A}}^{DDH}=|\Pr[\mathcal{A}(\varphi_t,\ \varphi_t^{\eta_1},\ \varphi_t^{\eta_2},\ \varphi_t^{\eta_1\cdot\eta_2})=1]-\Pr[\mathcal{A}(\varphi_t,\ \varphi_t^{\eta_1},\ \varphi_t^{\eta_2},\ \varphi_t^{r})=1]|$$

is negligible.

2.3. Decisional Bilinear Diffie–Hellman (DBDH) Assumption

Given $(\varphi,\ \varphi^{\eta_1},\ \varphi^{\eta_2},\ \varphi^{\eta_3})\in\mathbb{G}_1^4$ and $Z\in\mathbb{G}_T$, in which $(\eta_1,\ \eta_2,\ \eta_3)\in\mathbb{Z}_p^3$. $\mathcal{A}$'s aim is to determine whether $Z=\hat{e}(\varphi,\ \varphi)^{\eta_1\cdot\eta_2\cdot\eta_3}$ or $Z=\hat{e}(\varphi,\ \varphi)^r$, in which r is randomly selected from $\mathbb{Z}_p$. The DBDH assumption holds if $\mathcal{A}$'s advantage

$$\mathrm{Adv}_{\mathcal{A}}^{DBDH}=|\Pr[\mathcal{A}(\varphi,\ \varphi^{\eta_1},\ \varphi^{\eta_2},\ \varphi^{\eta_3},\ \hat{e}(\varphi,\ \varphi)^{\eta_1\cdot\eta_2\cdot\eta_3})=1]$$
$$-\Pr[\mathcal{A}(\varphi,\ \varphi^{\eta_1},\ \varphi^{\eta_2},\ \varphi^{\eta_3},\ \hat{e}(\varphi,\ \varphi)^{r})=1]|$$

is negligible.

3. Definition of CLPAEKS

3.1. System Model

The following three types of entities are involved in our CLPAEKS scheme.

- Key generation center (KGC): KGC generates the master secret key, the public parameters, and every user's partial secret key.
- Users: Include the sender and the receiver, which have been introduced in Section 1. Every user randomly selects a secret value and then generates its secret key using its partial secret key and the secret value.
- Cloud Server: It is a semi-trusted party managing the encrypted cloud database and responding to search requests.

3.2. Algorithms

The frequently used symbols are defined in Table 1. Our CLPAEKS scheme consists of the following algorithms.

Table 1. Notations.

Symbols	Meaning
λ	Security parameter
pp	Public parameters
msk	Master secret key
ID_i	A user's identity
$psk_i,\ x_i,\ sk_i,\ pk_i$	ID_i's partial secret key, secret value, secret key, and public key, respectively
$ID_s,\ pk_s, sk_s$	A sender's identity, public key, and secret key, respectively
$ID_r,\ pk_r,\ sk_r$	A receiver's identity, public key, and secret key, respectively
C	Searchable ciphertext
td	Trapdoor

1. Setup(λ): Run by KGC.
   - Input: λ.
   - Output: msk and pp.
2. Extract Partial Secret Key$(pp,\ msk,\ ID_i)$: Run by KGC.
   - Input: pp, msk, and ID_i.
   - Output: psk_i.
3. Extract Secret Value$(pp,\ ID_i)$: Run by the user ID_i.
   - Input: pp, ID_i.
   - Output: x_i.
4. Extract Secret Key$(pp,\ psk_i,\ x_i)$: Run by the user ID_i.
   - Input: pp, psk_i, x_i.
   - Output: sk_i.
5. Extract Public Key$(pp,\ x_i)$: Run by the user ID_i.
   - Input: pp, x_i.
   - Output: pk_i.
6. CLPAEKS$(pp,\ ID_s,\ sk_s,\ ID_r,\ pk_r,\ w)$: Run by the sender ID_s.
   - Input: pp, ID_s, sk_s, ID_r, pk_r, and a keyword w.
   - Output: C.
7. Trapdoor$(pp,\ ID_s,\ pk_s,\ ID_r,\ sk_r,\ w)$: Run by the receiver ID_r.
   - Input: pp, ID_s, pk_s, ID_r, sk_r, w.
   - Output: td.
8. Test$(C,\ td)$: Run by the cloud server.
   - Input: $C = \text{CLPAEKS}(pp,\ ID_s,\ sk_s,\ ID_r,\ pk_r,\ w)$ and $td = \text{Trapdoor}(pp,\ ID_s,\ pk_s,\ ID_r,\ sk_r,\ w')$.
   - Output: 1 will be output if $w = w'$, and 0 otherwise.

3.3. Security Model

The following two types of PPT adversaries are considered:

- Type-1 adversary: Denote this type of adversary with $\mathcal{A}_1$. $\mathcal{A}_1$ can replace any user's public key but cannot get the master secret key.
- Type-2 adversary: Denote this type of adversary with $\mathcal{A}_2$. $\mathcal{A}_2$ can get the master secret key but cannot replace any user's public key.

We consider two security properties, ciphertext indistinguishability and trapdoor indistinguishability. Since there are two types of adversaries in certificateless cryptosystems, we define the semantic security of CLPAEKS via four games. In Game $\mathcal{G}_1$ and Game $\mathcal{G}_2$, we formally define ciphertext indistinguishability in the same way as [23,24]. In Game $\mathcal{G}_3$ and Game $\mathcal{G}_4$, we formally define a stronger version of trapdoor indistinguishability. Different from [23,24], the adversary against trapdoor indistinguishability could freely access the trapdoor oracle in the games, which makes our definition of trapdoor indistinguishability stronger.

3.3.1. Game $\mathcal{G}_1$

1. Setup: The challenger $\mathcal{C}$ sends pp to $\mathcal{A}_1$.
2. Phase 1: $\mathcal{A}_1$ is allowed to access the following oracles.
   - $\mathcal{O}_{pk}(ID_i)$: Given ID_i, $\mathcal{C}$ returns pk_i.
   - $\mathcal{O}_{psk}(ID_i)$: Given ID_i, $\mathcal{C}$ returns psk_i.
   - $\mathcal{O}_{sk}(ID_i)$: Given ID_i, $\mathcal{C}$ returns sk_i. ID_i cannot occur in $\mathcal{O}_{sk}$ if ID_i's public key has been replaced.
   - $\mathcal{O}_{rpk}(ID_i,\ pk_i')$: Given ID_i and a new public key pk_i', $\mathcal{C}$ replaces pk_i with pk_i'.

- $\mathcal{O}_{CLPAEKS}(ID_s,\ ID_r,\ w)$: Given ID_s, ID_r and $w, \mathcal{C}$ returns $C \leftarrow \mathsf{CLPAEKS}(pp,\ ID_s,\ sk_s,\ ID_r,\ pk_r,\ w)$.
- $\mathcal{O}_T(ID_s,\ ID_r,\ w)$: Given ID_s, ID_r and $w, \mathcal{C}$ returns $td \leftarrow \mathsf{Trapdoor}(pp,\ ID_s,\ pk_s,\ ID_r,\ sk_r, w)$.

3. Challenge: $\mathcal{A}_1$ selects ID_{s^*}, ID_{r^*}, and two keywords (w_0^*, w_1^*) for the challenge, with the following restrictions: (1) Neither ID_{s^*} nor ID_{r^*} has been submitted to $\mathcal{O}_{psk}$; (2) Neither ID_{s^*} nor ID_{r^*} has been submitted to $\mathcal{O}_{sk}$; (3) Neither $(ID_{s^*}, ID_{r^*}, w_0^*)$ nor $(ID_{s^*}, ID_{r^*}, w_1^*)$ has been submitted to $\mathcal{O}_T$. $\mathcal{C}$ randomly selects $b \in \{0,1\}$ and sends $C^* \leftarrow \mathsf{CLPAEKS}(pp,\ ID_{s^*},\ sk_{s^*},\ ID_{r^*},\ pk_{r^*},\ w_b^*)$ to $\mathcal{A}_1$.
4. Phase 2: $\mathcal{A}_1$ is allowed to access the oracles as in Phase 1, with the following restrictions:
   - Neither ID_{s^*} nor ID_{r^*} can be submitted to $\mathcal{O}_{psk}$.
   - Neither ID_{s^*} nor ID_{r^*} can be submitted to $\mathcal{O}_{sk}$.
   - Neither $(ID_{s^*}, ID_{r^*}, w_0^*)$ nor $(ID_{s^*}, ID_{r^*}, w_1^*)$ can be submitted to $\mathcal{O}_T$.
5. Guess: $\mathcal{A}_1$ submits $b' \in \{0,1\}$. If $b = b'$, $\mathcal{A}_1$ wins the game. $\mathcal{A}_1$'s advantage is defined as

$$\mathrm{Adv}_{\mathcal{A}_1}^{CT-IND-CKA} = \left| Pr[b = b'] - \frac{1}{2} \right|.$$

Definition 1. *Our scheme satisfies* ***ciphertext indistinguishability under adaptive chosen-keyword attacks (CT-IND-CKA)*** *against Type-1 adversary if* $\mathrm{Adv}_{\mathcal{A}_1}^{CT-IND-CKA}$ *is negligible.*

3.3.2. Game $\mathcal{G}_2$

1. Setup: The challenger $\mathcal{C}$ sends pp and msk to $\mathcal{A}_2$.
2. Phase 1: $\mathcal{A}_2$ can is allowed to access the following oracles.
   - $\mathcal{O}_{pk}(ID_i)$: Same as $\mathcal{O}_{pk}$ in Game $\mathcal{G}_1$.
   - $\mathcal{O}_{psk}(ID_i)$: Same as $\mathcal{O}_{psk}$ in Game $\mathcal{G}_1$.
   - $\mathcal{O}_{sk}(ID_i)$: Given ID_i, $\mathcal{C}$ returns sk_i.
   - $\mathcal{O}_{CLPAEKS}(ID_s,\ ID_r,\ w)$: Same as $\mathcal{O}_{CLPAEKS}$ in Game $\mathcal{G}_1$.
   - $\mathcal{O}_T(ID_s,\ ID_r,\ w)$: Same as $\mathcal{O}_T$ in Game $\mathcal{G}_1$.
3. Challenge: $\mathcal{A}_2$ selects ID_{s^*}, ID_{r^*}, and two keywords (w_0^*, w_1^*) for challenge, with the following restrictions: (1) Neither ID_{s^*} nor ID_{r^*} has been submitted to $\mathcal{O}_{sk}$; (2) Neither $(ID_{s^*}, ID_{r^*}, w_0^*)$ nor $(ID_{s^*}, ID_{r^*}, w_1^*)$ has been submitted to $\mathcal{O}_T$. $\mathcal{C}$ randomly selects $b \in \{0,1\}$ and sends $C^* \leftarrow \mathsf{CLPAEKS}(pp,\ ID_{s^*},\ sk_{s^*},\ ID_{r^*},\ pk_{r^*},\ w_b^*)$ to $\mathcal{A}_2$.
4. Phase 2: $\mathcal{A}_2$ is allowed to access the oracles as in Phase 1, with the following restrictions:
   - Neither ID_{s^*} nor ID_{r^*} can be submitted to $\mathcal{O}_{sk}$.
   - Neither $(ID_{s^*}, ID_{r^*}, w_0^*)$ nor $(ID_{s^*}, ID_{r^*}, w_1^*)$ can be submitted to $\mathcal{O}_T$.
5. Guess: $\mathcal{A}_2$ submits $b' \in \{0,1\}$. If $b = b'$, $\mathcal{A}_2$ wins the game. $\mathcal{A}_2$'s advantage is defined as

$$\mathrm{Adv}_{\mathcal{A}_2}^{CT-IND-CKA} = \left| Pr[b = b'] - \frac{1}{2} \right|.$$

Definition 2. *Our scheme satisfies CT-IND-CKA against Type-2 adversary if* $\mathrm{Adv}_{\mathcal{A}_2}^{CT-IND-CKA}$ *is negligible.*

3.3.3. Game $\mathcal{G}_3$

1. Setup: The challenger $\mathcal{C}$ sends pp to $\mathcal{A}_1$.
2. Phase 1: Same as Phase 1 in Game $\mathcal{G}_1$.
3. Challenge: $\mathcal{A}_1$ selects ID_{s^*}, ID_{r^*}, and two keywords (w_0^*, w_1^*) for the challenge, with the following restrictions: (1) Neither ID_{s^*} nor ID_{r^*} has been submitted to $\mathcal{O}_{psk}$; (2) Neither ID_{s^*} nor ID_{r^*} has been submitted to $\mathcal{O}_{sk}$; (3) Neither $(ID_{s^*}, ID_{r^*}, w_0^*)$ nor $(ID_{s^*}, ID_{r^*}, w_1^*)$ has been submitted to $\mathcal{O}_{CLPAEKS}$. $\mathcal{C}$ randomly selects $b \in \{0,1\}$ and sends $td^* \leftarrow \mathsf{Trapdoor}(pp, ID_{s^*}, pk_{s^*}, ID_{r^*}, sk_{r^*}, w_b^*)$ to $\mathcal{A}_1$.

4. Phase 2: $\mathcal{A}_1$ is allowed to access the oracles as in Phase 1, with the following restrictions:
   - Neither ID_{s^*} nor ID_{r^*} can be submitted to $\mathcal{O}_{psk}$.
   - Neither ID_{s^*} nor ID_{r^*} can be submitted to $\mathcal{O}_{sk}$.
   - Neither $(ID_{s^*}, ID_{r^*}, w_0^*)$ nor $(ID_{s^*}, ID_{r^*}, w_1^*)$ can be submitted to $\mathcal{O}_{CLPAEKS}$.
5. Guess: $\mathcal{A}_1$ submits $b' \in \{0,1\}$. If $b = b'$, $\mathcal{A}_1$ wins the game. $\mathcal{A}_1$'s advantage is defined as

$$\mathrm{Adv}_{\mathcal{A}_1}^{S-TD-IND-CKA} = \left| Pr[b = b'] - \frac{1}{2} \right|.$$

Definition 3. *Our scheme satisfies* ***strong trapdoor indistinguishability under adaptive chosen-keyword attacks (S-TD-IND-CKA)*** *against Type-1 adversary if* $\mathrm{Adv}_{\mathcal{A}_1}^{S-TD-IND-CKA}$ *is negligible.*

3.3.4. Game $\mathcal{G}_4$

1. Setup: The challenger $\mathcal{C}$ sends pp and msk to $\mathcal{A}_2$.
2. Phase 1: Same as Phase 1 in Game $\mathcal{G}_2$.
3. Challenge: $\mathcal{A}_2$ selects ID_{s^*}, ID_{r^*}, and two keywords (w_0^*, w_1^*) for the challenge, with the following restrictions: (1) Neither ID_{s^*} nor ID_{r^*} has been submitted to $\mathcal{O}_{sk}$; (2) Neither $(ID_{s^*}, ID_{r^*}, w_0^*)$ nor $(ID_{s^*}, ID_{r^*}, w_1^*)$ has been submitted to $\mathcal{O}_{CLPAEKS}$. $\mathcal{C}$ randomly selects $b \in \{0,1\}$ and sends $td^* \leftarrow \mathsf{Trapdoor}(pp,\ ID_{s^*},\ pk_{s^*},\ ID_{r^*},\ sk_{r^*},\ w_b^*)$ to $\mathcal{A}_2$.
4. Phase 2: $\mathcal{A}_2$ is allowed to access the oracles as in Phase 1, with the following restrictions:
   - Neither ID_{s^*} nor ID_{r^*} can be submitted to $\mathcal{O}_{sk}$.
   - Neither $(ID_{s^*}, ID_{r^*}, w_0^*)$ nor $(ID_{s^*}, ID_{r^*}, w_1^*)$ can be submitted to $\mathcal{O}_{CLPAEKS}$.
5. Guess: $\mathcal{A}_2$ submits $b' \in \{0,1\}$. If $b = b'$, $\mathcal{A}_2$ wins the game. $\mathcal{A}_2$'s advantage is defined as

$$\mathrm{Adv}_{\mathcal{A}_2}^{S-TD-IND-CKA} = \left| Pr[b = b'] - \frac{1}{2} \right|.$$

Definition 4. *Our scheme satisfies S-TD-IND-CKA against Type-2 adversary if* $\mathrm{Adv}_{\mathcal{A}_2}^{S-TD-IND-CKA}$ *is negligible.*

4. The Proposed CLPAEKS Scheme

The frequently used symbols have been defined in Table 1. Following are the details of our CLPAEKS scheme.

1. $\mathsf{Setup}(\lambda)$: Run by KGC.
   - Input: Security parameter λ.
   - Select two cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_T$ with the same prime order $p > 2^\lambda$ and a bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$. Randomly select two generators $g \in \mathbb{G}_1$ and $g_t \in \mathbb{G}_T$.
   - Define 3 collision-resistant hash functions:
     - $H_1 : \{0,1\}^* \to \mathbb{G}_1$. It takes the user's identity as input.
     - $H_2 : \{0,1\}^* \to \mathbb{G}_1$. It takes the keyword as input.
     - $H_3 : \{0,1\}^* \times \{0,1\}^* \times \mathbb{G}_T \to \mathbb{Z}_p$.
   - Randomly select $y \in \mathbb{Z}_p$. Set master secret key $msk = y$ and master public key $mpk = g^y$.
   - Output: $pp = \{p,\ \mathbb{G}_1,\ \mathbb{G}_T,\ \hat{e},\ g,\ g_t,\ H_1,\ H_2,\ H_3,\ mpk\}$.
2. $\mathsf{Extract\ Partial\ Secret\ Key}(pp,\ msk,\ ID_i)$: Run by KGC.
   - Input: pp, msk, and a user's identity ID_i.
   - Output: ID_i's partial secret key $psk_i = H_1(ID_i)^y$.
3. $\mathsf{Extract\ Secret\ Value}(pp,\ ID_i)$: Run by the user ID_i.
   - Input: pp, ID_i.

- Output: ID_i's secret value x_i, which is randomly selected from $\mathbb{Z}_p$.

4. Extract Secret Key$(pp,\ psk_i,\ x_i)$: Run by the user ID_i.
   - Input: pp, psk_i, x_i.
   - Output: ID_i's secret key $sk_i = (sk_{i,1},\ sk_{i,2}) = (x_i,\ psk_i)$.
5. Extract Public Key$(pp,\ x_i)$: Run by the user ID_i.
   - Input: pp, x_i.
   - Output: ID_i's public key $pk_i = g_t^{x_i}$.
6. CLPAEKS$(pp,\ ID_s,\ sk_s,\ ID_r,\ pk_r,\ w)$: Run by the sender ID_s.
   - Input: pp, ID_s, $sk_s = (sk_{s,1},\ sk_{s,2}) = (x_s,\ H_1(ID_s)^y)$, ID_r, $pk_r = g_t^{x_r}$, and a keyword w.
   - Randomly select $\alpha \in \mathbb{Z}_p$.
   - Compute $C = (c_1,\ c_2,\ c_3)$:

   $$c_1 = \hat{e}(g,\ H_2(w))^{\alpha \cdot k}, \quad c_2 = g^{\alpha}, \quad c_3 = g^{\frac{\alpha}{k}},$$

   in which

   $$\begin{aligned} k &= H_3(ID_s \parallel ID_r \parallel pk_r^{sk_{s,1}} \cdot \hat{e}(sk_{s,2},\ H_1(ID_r))) \\ &= H_3(ID_s \parallel ID_r \parallel g_t^{x_s \cdot x_r} \cdot \hat{e}(H_1(ID_s), H_1(ID_r))^y). \end{aligned}$$

   - Output: $C = (c_1,\ c_2,\ c_3)$.
7. Trapdoor$(pp, ID_s, pk_s, ID_r, sk_r, w)$: Run by the receiver ID_r.
   - Input: pp, ID_s, $pk_s = g_t^{x_s}$, ID_r, $sk_r = (sk_{r,1},\ sk_{r,2}) = (x_r,\ H_1(ID_r)^y)$, and a keyword w.
   - Randomly select $(\beta, \gamma) \in \mathbb{Z}_p^2$.
   - Compute $td = (td_1,\ td_2,\ td_3)$:

   $$td_1 = H_2(w)^{\beta + \frac{\gamma}{k}}, \quad td_2 = H_2(w)^{\frac{k^3}{\beta} - \gamma}, \quad td_3 = \frac{\beta}{k} + \frac{k}{\beta},$$

   in which

   $$\begin{aligned} k &= H_3(ID_s \parallel ID_r \parallel pk_s^{sk_{r,1}} \cdot \hat{e}(sk_{r,2},\ H_1(ID_s))) \\ &= H_3(ID_s \parallel ID_r \parallel g_t^{x_s \cdot x_r} \cdot \hat{e}(H_1(ID_s),\ H_1(ID_r))^y). \end{aligned}$$

   - Output: $td = (td_1,\ td_2,\ td_3)$.
8. Test$(C,\ td)$: Run by the cloud server.
   - Input: $C = (c_1,\ c_2,\ c_3)$ and $td = (td_1, td_2, td_3)$.
   - Output: Check whether

   $$c_1^{td_3} = \hat{e}(c_2,\ td_1) \cdot \hat{e}(c_3,\ td_2)$$

   holds, if it holds then output 1, and 0 otherwise.

5. Security Analysis

5.1. CT-IND-CKA against Type-1 Adversary

Theorem 1. *Our scheme satisfies CT-IND-CKA against Type-1 adversary in the random oracle model if the DBDH assumption holds.*

Proof. Suppose that $\mathrm{Adv}_{\mathcal{A}_1}^{CT-IND-CKA} = \epsilon$. Given a DBDH instance $(\mathbb{G}_1,\ \mathbb{G}_T,\ \hat{e},\ g,\ g^{\eta_1},\ g^{\eta_2},\ g^{\eta_3},\ Z)$. Denoted by $\zeta = 0$ that $Z = \hat{e}(g,\ g)^{\eta_1 \cdot \eta_2 \cdot \eta_3}$, and by $\zeta = 1$ that Z is random. In the following, we construct a simulator $\mathcal{B}$ that runs $\mathcal{A}_1$ as a subroutine to correctly guess the value of ζ.

1. Setup: $\mathcal{B}$ sets $mpk = g^{\eta_1}$, implying that $msk = \eta_1$, in which η_1 is unknown to $\mathcal{B}$. Then sends pp to $\mathcal{A}_1$.
2. Phase 1: $\mathcal{A}_1$ is allowed to access the following oracles.
   - $\mathcal{O}_{H_1}(ID_i)$: Suppose that there are q_{H_1} distinct queries to $\mathcal{O}_{H_1}$. $\mathcal{B}$ randomly selects $(i^*, j^*) \in \{1, \cdots, q_{H_1}\}$ as its guess of the identities selected by $\mathcal{A}_1$ for challenge. For ID_i:
     - If $i = i^*$, $\mathcal{B}$ adds $\{ID_{i^*}, -, g^{\eta_2}\}$ to list L_{H_1} and returns g^{η_2} to $\mathcal{A}_1$.
     - If $i = j^*$, $\mathcal{B}$ adds $\{ID_{j^*}, -, g^{\eta_3}\}$ to list L_{H_1} and returns g^{η_3} to $\mathcal{A}_1$.
     - Otherwise, $\mathcal{B}$ randomly selects $h_{1,i} \in \mathbb{Z}_p$, adds $\left\{ID_i, h_{1,i}, g^{h_{1,i}}\right\}$ to list L_{H_1}, and returns $g^{h_{1,i}}$ to $\mathcal{A}_1$.

     If the repeated queries are submitted, the answer that already exists in L_{H_1} will be returned.
   - $\mathcal{O}_{H_2}$: Given $w \in \{0,1\}^*$, $\mathcal{B}$ randomly selects $h_2 \in \mathbb{G}_1$, adds $\{w, h_2\}$ to list L_{H_2}, and returns h_2. If the repeated queries are submitted, the answer that already exists in L_{H_2} will be returned.
   - $\mathcal{O}_{H_3}$: Given $(u_1, u_2, u_3) \in \{0,1\}^* \times \{0,1\}^* \times \mathbb{G}_T$. $\mathcal{B}$ randomly selects $h_3 \in \mathbb{Z}_p$, adds $\{u_1, u_2, u_3, h_3\}$ to list L_{H_3}, and returns h_3. If the repeated queries are submitted, the answer that already exists in L_{H_3} will be returned.
   - $\mathcal{O}_{pk}(ID_i)$: $\mathcal{B}$ randomly selects $x_i \in \mathbb{Z}_p$, then:
     - If $i \neq i^* \wedge i \neq j^*$, $\mathcal{B}$ calls $\mathcal{O}_{H_1}(ID_i)$, retrieves $\left\{ID_i, h_{1,i}, g^{h_{1,i}}\right\}$ from L_{H_1}, sets

       $$pk_i = g_t^{x_i}, \quad psk_i = g^{\eta_1 \cdot h_{1,i}},$$

       adds $\{ID_i, pk_i, psk_i, x_i\}$ to list L_{key}, and returns pk_i.
     - Otherwise, $\mathcal{B}$ calls $\mathcal{O}_{H_1(ID_i)}$ and sets

       $$pk_i = g_t^{x_i},$$

       adds $\{ID_i, pk_i, -, x_i\}$ to list L_{key}, and returns pk_i.

     If the repeated queries are submitted, the answer that already exists in L_{key} will be returned.
   - $\mathcal{O}_{psk}(ID_i)$:
     - If $i = i^* \vee i = j^*$, $\mathcal{B}$ aborts.
     - Otherwise, $\mathcal{B}$ calls $\mathcal{O}_{pk}(ID_i)$, retrieves $\{ID_i, pk_i, psk_i, x_i\}$ from L_{key}, and returns psk_i.
   - $\mathcal{O}_{sk}(ID_i)$:
     - If $i = i^* \vee i = j^*$, $\mathcal{B}$ aborts.
     - Otherwise, $\mathcal{B}$ calls $\mathcal{O}_{pk}(ID_i)$, retrieves $\{ID_i, pk_i, psk_i, x_i\}$ from L_{key}, and returns $sk_i = (psk_i, x_i)$.

     ID_i cannot occur in $\mathcal{O}_{sk}$ if ID_i's public key has been replaced.
   - $\mathcal{O}_{rpk}(ID_i, pk_i')$: $\mathcal{B}$ calls $\mathcal{O}_{pk}(ID_i)$ and replaces $\{ID_i, pk_i, psk_i, x_i\}$ with $\{ID_i, pk_i', psk_i, -\}$.
   - $\mathcal{O}_{CLPAEKS}(ID_s, ID_r, w)$: $\mathcal{B}$ randomly selects $\alpha \in \mathbb{Z}_p$ and returns $C = (c_1, c_2, c_3)$:

     $$c_1 = \hat{e}(g, H_2(w))^{\alpha \cdot k}, \quad c_2 = g^{\alpha}, \quad c_3 = g^{\frac{\alpha}{k}},$$

     in which k is different based on the following cases.
     - If $s = i^* \wedge r = j^*$, $k = H_3(ID_{i^*} \parallel ID_{j^*} \parallel g_t^{x_{i^*} \cdot x_{j^*}} \cdot Z)$.

- If $s = j^* \wedge r = i^*$, $k = H_3(ID_{j^*} \parallel ID_{i^*} \parallel g_t^{x_{i^*} \cdot x_{j^*}} \cdot Z)$.
- Otherwise, it means that $(s \neq i^* \wedge s \neq j^*) \vee (r \neq i^* \wedge r \neq j^*)$.
  * If $s \neq i^* \wedge s \neq j^*$, $\mathcal{B}$ retrieves $\left\{ID_s,\ h_{1,s},\ g^{h_{1,s}}\right\}$ from L_{H_1} and computes $k = H_3(ID_s \parallel ID_r \parallel g_t^{x_s \cdot x_r} \cdot \hat{e}(g^{\eta_1},\ H_1(ID_r))^{h_{1,s}})$.
  * Otherwise, $\mathcal{B}$ retrieves $\left\{ID_r,\ h_{1,r},\ g^{h_{1,r}}\right\}$ from L_{H_1} and computes $k = H_3(ID_s \parallel ID_r \parallel g_t^{x_s \cdot x_r} \cdot \hat{e}(g^{\eta_1},\ H_1(ID_s))^{h_{1,r}})$.

- $\mathcal{O}_T(ID_s,\ ID_r,\ w)$: $\mathcal{B}$ randomly selects $(\beta, \gamma) \in \mathbb{Z}_p^2$ and returns $td = (td_1,\ td_2,\ td_3)$:

$$td_1 = H_2(w)^{\beta + \frac{\gamma}{k}}, \quad td_2 = H_2(w)^{\frac{k^3}{\beta} - \gamma}, \quad td_3 = \frac{\beta}{k} + \frac{k}{\beta},$$

  in which k is different based on the following cases.
  - If $s = i^* \wedge r = j^*$, $k = H_3(ID_{i^*} \parallel ID_{j^*} \parallel g_t^{x_{i^*} \cdot x_{j^*}} \cdot Z)$.
  - If $s = j^* \wedge r = i^*$, $k = H_3(ID_{j^*} \parallel ID_{i^*} \parallel g_t^{x_{i^*} \cdot x_{j^*}} \cdot Z)$.
  - Otherwise, it means that $(s \neq i^* \wedge s \neq j^*) \vee (r \neq i^* \wedge r \neq j^*)$.
    * If $s \neq i^* \wedge s \neq j^*$, $\mathcal{B}$ retrieves $\left\{ID_s,\ h_{1,s},\ g^{h_{1,s}}\right\}$ from L_{H_1} and computes $k = H_3(ID_s \parallel ID_r \parallel g_t^{x_s \cdot x_r} \cdot \hat{e}(g^{\eta_1},\ H_1(ID_r))^{h_{1,s}})$.
    * Otherwise, $\mathcal{B}$ retrieves $\left\{ID_r,\ h_{1,r},\ g^{h_{1,r}}\right\}$ from L_{H_1} and computes $k = H_3(ID_s \parallel ID_r \parallel g_t^{x_s \cdot x_r} \cdot \hat{e}(g^{\eta_1},\ H_1(ID_s))^{h_{1,r}})$.

3. Challenge: $\mathcal{A}_1$ selects ID_{s^*}, ID_{r^*}, and two keywords (w_0^*, w_1^*) for the challenge, with the following restrictions: (1) Neither ID_{s^*} nor ID_{r^*} has been submitted to $\mathcal{O}_{psk}$; (2) Neither ID_{s^*} nor ID_{r^*} has been submitted to $\mathcal{O}_{sk}$; (3) Neither $(ID_{s^*},\ ID_{r^*},\ w_0^*)$ nor $(ID_{s^*},\ ID_{r^*},\ w_1^*)$ has been submitted to $\mathcal{O}_T$. If $\neg(s^* = i^* \wedge r^* = j^*) \wedge \neg(s^* = j^* \wedge r^* = i^*)$, $\mathcal{B}$ aborts and randomly returns $\zeta' \in \{0,1\}$. Otherwise, $\mathcal{B}$ randomly selects $b \in \{0,1\}$ and sends $C^* = (c_1^*,\ c_2^*,\ c_3^*)$ to $\mathcal{A}_1$, in which

$$\alpha^* \in \mathbb{Z}_p, \quad k^* = H_3(ID_{s^*} \parallel ID_{r^*} \parallel g_t^{x_{s^*} \cdot x_{r^*}} \cdot Z),$$

$$c_1^* = \hat{e}(g,\ H_2(w_b^*))^{\alpha^* \cdot k^*}, \quad c_2 = g^{\alpha^*}, \quad c_3 = g^{\frac{\alpha^*}{k^*}}.$$

4. Phase 2: $\mathcal{A}_1$ is allowed to access the oracles as in Phase 1, with the following restrictions:
   - Neither ID_{s^*} nor ID_{r^*} can be submitted to $\mathcal{O}_{psk}$.
   - Neither ID_{s^*} nor ID_{r^*} can be submitted to $\mathcal{O}_{sk}$.
   - Neither $(ID_{s^*},\ ID_{r^*},\ w_0^*)$ nor $(ID_{s^*},\ ID_{r^*},\ w_1^*)$ can be submitted to $\mathcal{O}_T$.
5. Guess: $\mathcal{A}_1$ submits b'. If $b = b'$, $\mathcal{A}_1$ wins, and $\mathcal{B}$ returns $\zeta' = 0$. Otherwise, $\mathcal{A}_1$ loses, and $\mathcal{B}$ returns $\zeta' = 1$.

If $\zeta = 0$, $\mathcal{B}$ perfectly simulates Section 3.3.1, and $\mathcal{A}_1$'s probability of winning is $\epsilon + 1/2$. Otherwise, C^* is independent of w_b^*, and $\mathcal{A}_1$'s probability of winning is $1/2$. $\mathcal{B}$ aborts and randomly returns $\zeta' \in \{0,1\}$ if its guess of the identities selected by $\mathcal{A}_1$ for challenge is wrong. Denote $\mathcal{B}$'s abortion with abt, we have

$$\Pr[\zeta' = \zeta | abt] = \frac{1}{2},$$

$$\Pr[\zeta' = \zeta | \overline{abt}] = (\epsilon + \frac{1}{2}) \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{\epsilon}{2} + \frac{1}{2},$$

$$\Pr[\overline{abt}] \geq \frac{1}{\mathrm{C}_{q_{H_1}}^2} = \frac{2}{q_{H_1}(q_{H_1} - 1)}.$$

$\mathcal{B}$'s advantage in solving the DBDH problem is

$$\begin{aligned}
&\mathrm{Adv}^{DBDH} \\
&= \left|\Pr[\zeta' = \zeta] - \frac{1}{2}\right| \\
&= \left|\Pr[\zeta' = \zeta \wedge \overline{abt}] + \Pr[\zeta' = \zeta \wedge abt] - \frac{1}{2}\right| \\
&= \left|\Pr[\zeta' = \zeta|\overline{abt}] \cdot \Pr[\overline{abt}] + \Pr[\zeta' = \zeta|abt] \cdot \Pr[abt] - \frac{1}{2}\right| \\
&= \left|(\frac{\epsilon}{2} + \frac{1}{2}) \cdot \Pr[\overline{abt}] + \frac{1}{2} \cdot (1 - \Pr[\overline{abt}]) - \frac{1}{2}\right| \\
&= \frac{\epsilon}{2} \cdot \Pr[\overline{abt}] \\
&\geq \frac{\epsilon}{q_{H_1}(q_{H_1} - 1)}.
\end{aligned}$$

ϵ is negligible due to the intractability of the DBDH problem. This completes the proof. □

5.2. CT-IND-CKA against Type-2 Adversary

Theorem 2. *Our scheme satisfies CT-IND-CKA against Type-2 adversary in the random oracle model if the DDH assumption in $\mathbb{G}_T$ holds.*

Proof. Suppose that $\mathrm{Adv}_{\mathcal{A}_2}^{CT-IND-CKA} = \epsilon$. Given a DDH instance $(g_t, g_t^{\eta_1}, g_t^{\eta_2}, Z) \in \mathbb{G}_T^4$. Denoted by $\zeta = 0$ that $Z = g_t^{\eta_1 \cdot \eta_2}$, and by $\zeta = 1$ that Z is random. In the following, we construct a simulator $\mathcal{B}$ that runs $\mathcal{A}_2$ as a subroutine to correctly guess the value of ζ.

1. Setup: $\mathcal{B}$ sends pp and $msk = y$ to $\mathcal{A}_2$.
2. Phase 1: $\mathcal{A}_2$ is allowed to access the following oracles:
   - $\mathcal{O}_{H_1}(ID_i)$: $\mathcal{B}$ randomly selects $h_{1,i} \in \mathbb{Z}_p$, adds $\left\{ID_i,\ h_{1,i},\ g^{h_{1,i}}\right\}$ to list L_{H_1}, and returns $g^{h_{1,i}}$. If the repeated queries are submitted, the answer that already exists in L_{H_1} will be returned. Suppose that there are q_{H_1} distinct queries to $\mathcal{O}_{H_1}$. $\mathcal{B}$ randomly selects $(i^*,\ j^*) \in \{1, \cdots, q_{H_1}\}$ as its guess of the identities selected by $\mathcal{A}_1$ for challenge.
   - $\mathcal{O}_{H_2}$: Same as $\mathcal{O}_{H_2}$ in the proof of Theorem 1.
   - $\mathcal{O}_{H_3}$: Same as $\mathcal{O}_{H_3}$ in the proof of Theorem 1.
   - $\mathcal{O}_{pk}(ID_i)$: $\mathcal{B}$ calls $\mathcal{O}_{H_1}(ID_i)$ and retrieves $\left\{ID_i,\ h_{1,i},\ g^{h_{1,i}}\right\}$ from L_{H_1}, then:
     – If $i = i^*$, $\mathcal{B}$ sets
     $$pk_i = g_t^{\eta_1}, \quad psk_i = g^{y \cdot h_{1,i}},$$
     adds $\{ID_i,\ pk_i,\ psk_i,\ -\}$ to list L_{key}, and returns pk_i to $\mathcal{A}_2$.
     – If $i = j^*$, $\mathcal{B}$ sets
     $$pk_i = g_t^{\eta_2}, \quad psk_i = g^{y \cdot h_{1,i}},$$
     adds $\{ID_i,\ pk_i,\ psk_i,\ -\}$ to list L_{key}, and returns pk_i to $\mathcal{A}_2$.
     – Otherwise, $\mathcal{B}$ randomly selects $x_i \in \mathbb{Z}_p$, sets
     $$pk_i = g_t^{x_i}, \quad psk_i = g^{y \cdot h_{1,i}},$$
     adds $\{ID_i,\ pk_i,\ psk_i,\ x_i\}$ to list L_{key}, and returns pk_i to $\mathcal{A}_2$.

   If the repeated queries are submitted, the answer that already exists in L_{key} will be returned.

- $\mathcal{O}_{psk}(ID_i)$: $\mathcal{B}$ calls $\mathcal{O}_{pk}(ID_i)$, retrieves $\{ID_i,\ pk_i,\ psk_i,\ x_i\}$ from L_{key}, and returns psk_i to $\mathcal{A}_2$.
- $\mathcal{O}_{sk}(ID_i)$:
  - If $i = i^* \vee i = j^*$, $\mathcal{B}$ aborts.
  - Otherwise, $\mathcal{B}$ calls $\mathcal{O}_{pk}(ID_i)$, retrieves $\{ID_i,\ pk_i,\ psk_i,\ x_i\}$ from L_{key}, and returns $sk_i = (psk_i, x_i)$.
- $\mathcal{O}_{CLPAEKS}(ID_s,\ ID_r,\ w)$: $\mathcal{B}$ randomly selects $\alpha \in \mathbb{Z}_p$ and returns $C = (c_1,\ c_2,\ c_3)$:

$$c_1 = \hat{e}(g,\ H_2(w))^{\alpha \cdot k}, \quad c_2 = g^{\alpha}, \quad c_3 = g^{\frac{\alpha}{k}},$$

  in which k is different based on the following cases.
  - If $s = i^* \wedge r = j^*$, $k = H_3(ID_{i^*} \parallel ID_{j^*} \parallel Z \cdot \hat{e}(H_1(ID_{i^*}),\ H_1(ID_{j^*}))^y)$.
  - If $s = j^* \wedge r = i^*$, $k = H_3(ID_{j^*} \parallel ID_{i^*} \parallel Z \cdot \hat{e}(H_1(ID_{i^*}),\ H_1(ID_{j^*}))^y)$.
  - Otherwise, it means that $(s \neq i^* \wedge s \neq j^*) \vee (r \neq i^* \wedge r \neq j^*)$.
    * If $s \neq i^* \wedge s \neq j^*$, $\mathcal{B}$ retrieves $\{ID_s,\ pk_s,\ psk_s,\ x_s\}$ from L_{key} and computes $k = H_3(ID_s \parallel ID_r \parallel pk_r^{x_s} \cdot \hat{e}(H_1(ID_s),\ H_1(ID_r))^y)$.
    * Otherwise, $\mathcal{B}$ retrieves $\{ID_r,\ pk_r,\ psk_r,\ x_r\}$ from L_{key} and computes $k = H_3(ID_s \parallel ID_r \parallel pk_s^{x_r} \cdot \hat{e}(H_1(ID_s),\ H_1(ID_r))^y)$.
- $\mathcal{O}_T(ID_s, ID_r, w)$: $\mathcal{B}$ randomly selects $(\beta, \gamma) \in \mathbb{Z}_p^2$ and returns $td = (td_1,\ td_2,\ td_3)$:

$$td_1 = H_2(w)^{\beta + \frac{\gamma}{k}}, \quad td_2 = H_2(w)^{\frac{k^3}{\beta} - \gamma}, \quad td_3 = \frac{\beta}{k} + \frac{k}{\beta},$$

  in which k is different based on the following cases.
  - If $s = i^* \wedge r = j^*$, $k = H_3(ID_{i^*} \parallel ID_{j^*} \parallel Z \cdot \hat{e}(H_1(ID_{i^*}),\ H_1(ID_{j^*}))^y)$.
  - If $s = j^* \wedge r = i^*$, $k = H_3(ID_{j^*} \parallel ID_{i^*} \parallel Z \cdot \hat{e}(H_1(ID_{i^*}),\ H_1(ID_{j^*}))^y)$.
  - Otherwise, it means that $(s \neq i^* \wedge s \neq j^*) \vee (r \neq i^* \wedge r \neq j^*)$.
    * If $s \neq i^* \wedge s \neq j^*$, $\mathcal{B}$ retrieves $\{ID_s,\ pk_s,\ psk_s,\ x_s\}$ from L_{key} and computes $k = H_3(ID_s \parallel ID_r \parallel pk_r^{x_s} \cdot \hat{e}(H_1(ID_s),\ H_1(ID_r))^y)$.
    * Otherwise, $\mathcal{B}$ retrieves $\{ID_r,\ pk_r,\ psk_r,\ x_r\}$ from L_{key} and computes $k = H_3(ID_s \parallel ID_r \parallel pk_s^{x_r} \cdot \hat{e}(H_1(ID_s),\ H_1(ID_r))^y)$.

3. Challenge: $\mathcal{A}_2$ selects ID_{s^*}, ID_{r^*}, and two keywords $(w_0^*,\ w_1^*)$ for the challenge, with the following restrictions: (1) Neither ID_{s^*} nor ID_{r^*} has been submitted to $\mathcal{O}_{sk}$; (2) Neither $(ID_{s^*},\ ID_{r^*},\ w_0^*)$ nor $(ID_{s^*},\ ID_{r^*},\ w_1^*)$ has been submitted to $\mathcal{O}_T$. If $\neg(s^* = i^* \wedge r^* = j^*) \wedge \neg(s^* = j^* \wedge r^* = i^*)$, $\mathcal{B}$ aborts and randomly returns $\zeta' \in \{0,1\}$. Otherwise, $\mathcal{B}$ randomly selects $b \in \{0,1\}$ and sends $C^* = (c_1^*,\ c_2^*,\ c_3^*)$ to $\mathcal{A}_2$, in which

$$\alpha^* \in \mathbb{Z}_p, \quad k^* = H_3(ID_{s^*} \parallel ID_{r^*} \parallel Z \cdot \hat{e}(H_1(ID_{s^*}),\ H_1(ID_{r^*}))^y),$$

$$c_1^* = \hat{e}(g,\ H_2(w_b^*))^{\alpha^* \cdot k^*}, \quad c_2 = g^{\alpha^*}, \quad c_3 = g^{\frac{\alpha^*}{k^*}}.$$

4. Phase 2: $\mathcal{A}_2$ is allowed to access the oracles as in Phase 1, with the following restrictions:
   - Neither ID_{s^*} nor ID_{r^*} can be submitted to $\mathcal{O}_{sk}$.
   - Neither $(ID_{s^*}, ID_{r^*}, w_0^*)$ nor $(ID_{s^*}, ID_{r^*}, w_1^*)$ can be submitted to $\mathcal{O}_T$.
5. Guess: $\mathcal{A}_2$ submits b'. If $b = b'$, $\mathcal{A}_2$ wins, and $\mathcal{B}$ returns $\zeta' = 0$. If $b \neq b'$, $\mathcal{A}_2$ loses, and $\mathcal{B}$ returns $\zeta' = 1$.

If $\zeta = 0$, $\mathcal{B}$ perfectly simulates Section 3.3.2, and $\mathcal{A}_2$'s probability of winning is $\epsilon + 1/2$. Otherwise, C^* is independent of w_b^*, and $\mathcal{A}_2$'s probability of winning is $1/2$. $\mathcal{B}$ aborts and randomly returns $\zeta' \in \{0,1\}$ if its guess of the identities selected by $\mathcal{A}_2$ for challenge is wrong. Denote $\mathcal{B}$'s abortion with abt, we have

$$\Pr[\zeta' = \zeta | abt] = \frac{1}{2},$$

$$\Pr[\zeta' = \zeta|\overline{abt}] = (\epsilon + \frac{1}{2}) \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{\epsilon}{2} + \frac{1}{2},$$
$$\Pr[\overline{abt}] \geq \frac{1}{C^2_{q_{H_1}}} = \frac{2}{q_{H_1}(q_{H_1} - 1)}.$$

$\mathcal{B}$'s advantage in solving the DDH problem in $\mathbb{G}_T$ is

$$\begin{aligned}
&\mathrm{Adv}^{DDH} \\
&= \left|\Pr[\zeta' = \zeta] - \frac{1}{2}\right| \\
&= \left|\Pr[\zeta' = \zeta \wedge \overline{abt}] + \Pr[\zeta' = \zeta \wedge abt] - \frac{1}{2}\right| \\
&= \left|\Pr[\zeta' = \zeta|\overline{abt}] \cdot \Pr[\overline{abt}] + \Pr[\zeta' = \zeta|abt] \cdot \Pr[abt] - \frac{1}{2}\right| \\
&= \left|(\frac{\epsilon}{2} + \frac{1}{2}) \cdot \Pr[\overline{abt}] + \frac{1}{2} \cdot (1 - \Pr[\overline{abt}]) - \frac{1}{2}\right| \\
&= \frac{\epsilon}{2} \cdot \Pr[\overline{abt}] \\
&\geq \frac{\epsilon}{q_{H_1}(q_{H_1} - 1)}.
\end{aligned}$$

ϵ is negligible due to the intractability of the DDH problem in $\mathbb{G}_T$. This completes the proof. □

5.3. S-TD-IND-CKA against Type-1 Adversary

Theorem 3. *Our scheme satisfies S-TD-IND-CKA against Type-1 adversary in the random oracle model if the DBDH assumption holds.*

Proof. Suppose that $\mathrm{Adv}^{S-TD-IND-CKA}_{\mathcal{A}_1} = \epsilon$. Given a DBDH instance $(\mathbb{G}_1, \mathbb{G}_T, \hat{e}, g, g^{\eta_1}, g^{\eta_2}, g^{\eta_3}, Z)$. Denoted by $\zeta = 0$ that $Z = \hat{e}(g, g)^{\eta_1 \cdot \eta_2 \cdot \eta_3}$, and by $\zeta = 1$ that Z is random. In the following, we construct a simulator $\mathcal{B}$ that runs $\mathcal{A}_1$ as a subroutine to correctly guess the value of ζ.

1. Setup: $\mathcal{B}$ sets $mpk = g^{\eta_1}$, implying that $msk = \eta_1$, in which η_1 is unknown to $\mathcal{B}$. Then sends pp to $\mathcal{A}_1$.
2. Phase 1: Same as Phase 1 in the proof of Theorem 1.
3. Challenge: $\mathcal{A}_1$ selects ID_{s^*}, ID_{r^*}, and two keywords (w_0^*, w_1^*) for the challenge, with the following restrictions: (1) Neither ID_{s^*} nor ID_{r^*} has been submitted to $\mathcal{O}_{psk}$; (2) Neither ID_{s^*} nor ID_{r^*} has been submitted to $\mathcal{O}_{sk}$; (3) Neither $(ID_{s^*}, ID_{r^*}, w_0^*)$ nor $(ID_{s^*}, ID_{r^*}, w_1^*)$ has been submitted to $\mathcal{O}_{CLPAEKS}$. If $\neg(s^* = i^* \wedge r^* = j^*) \wedge \neg(s^* = j^* \wedge r^* = i^*)$, $\mathcal{B}$ aborts and randomly returns $\zeta' \in \{0, 1\}$. Otherwise, $\mathcal{B}$ randomly selects $b \in \{0, 1\}$ and sends $td^* = (td_1^*, td_2^*, td_3^*)$ to $\mathcal{A}_1$, in which
$$(\beta^*, \gamma^*) \in \mathbb{Z}_p^2, \quad k^* = H_3(ID_{s^*} \parallel ID_{r^*} \parallel g_t^{x_{s^*} \cdot x_{r^*}} \cdot Z),$$
$$td_1^* = H_2(w_b^*)^{\beta^* + \frac{\gamma^*}{k^*}}, \quad td_2^* = H_2(w_b^*)^{\frac{(k^*)^3}{\beta^*} - \gamma^*}, \quad td_3^* = \frac{\beta^*}{k^*} + \frac{k^*}{\beta^*}.$$
4. Phase 2: $\mathcal{A}_1$ is allowed to access the oracles as in Phase 1, with the following restrictions:
   - Neither ID_{s^*} nor ID_{r^*} can be submitted to $\mathcal{O}_{psk}$.
   - Neither ID_{s^*} nor ID_{r^*} can be submitted to $\mathcal{O}_{sk}$.
   - Neither $(ID_{s^*}, ID_{r^*}, w_0^*)$ nor $(ID_{s^*}, ID_{r^*}, w_1^*)$ can be submitted to $\mathcal{O}_{CLPAEKS}$.
5. Guess: $\mathcal{A}_1$ submits b'. If $b = b'$, $\mathcal{A}_1$ wins, and $\mathcal{B}$ returns $\zeta' = 0$. Otherwise, $\mathcal{A}_1$ loses, and $\mathcal{B}$ returns $\zeta' = 1$.

If $\zeta = 0$, $\mathcal{B}$ perfectly simulates Section 3.3.3, and $\mathcal{A}_1$'s probability of winning is $\epsilon + 1/2$. Otherwise, td^* is independent of w_b^*, and $\mathcal{A}_1$'s probability of winning is $1/2$. $\mathcal{B}$ aborts and randomly returns $\zeta' \in \{0,1\}$ if its guess of the identities selected by $\mathcal{A}_1$ for challenge is wrong. Denote $\mathcal{B}$'s abortion with abt, we have

$$\Pr[\zeta' = \zeta | abt] = \frac{1}{2},$$

$$\Pr[\zeta' = \zeta | \overline{abt}] = (\epsilon + \frac{1}{2}) \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{\epsilon}{2} + \frac{1}{2},$$

$$\Pr[\overline{abt}] \geq \frac{1}{C^2_{q_{H_1}}} = \frac{2}{q_{H_1}(q_{H_1} - 1)}.$$

$\mathcal{B}$'s advantage in solving the DBDH problem is

$$\begin{aligned}
&\mathrm{Adv}^{DBDH}\\
&= \left|\Pr[\zeta' = \zeta] - \frac{1}{2}\right|\\
&= \left|\Pr[\zeta' = \zeta \wedge \overline{abt}] + \Pr[\zeta' = \zeta \wedge abt] - \frac{1}{2}\right|\\
&= \left|\Pr[\zeta' = \zeta | \overline{abt}] \cdot \Pr[\overline{abt}] + \Pr[\zeta' = \zeta | abt] \cdot \Pr[abt] - \frac{1}{2}\right|\\
&= \left|(\frac{\epsilon}{2} + \frac{1}{2}) \cdot \Pr[\overline{abt}] + \frac{1}{2} \cdot (1 - \Pr[\overline{abt}]) - \frac{1}{2}\right|\\
&= \frac{\epsilon}{2} \cdot \Pr[\overline{abt}]\\
&\geq \frac{\epsilon}{q_{H_1}(q_{H_1} - 1)}.
\end{aligned}$$

ϵ is negligible due to the intractability of the DDH problem. This completes the proof. □

5.4. S-TD-IND-CKA against Type-2 Adversary

Theorem 4. *Our scheme satisfies S-TD-IND-CKA against Type-2 adversary in the random oracle model if the DDH assumption in $\mathbb{G}_T$ holds.*

Proof. Suppose that $\mathrm{Adv}_{\mathcal{A}_2}^{S-TD-IND-CKA} = \epsilon$. Given a DDH instance $(g_t, g_t^{\eta_1}, g_t^{\eta_2}, Z) \in \mathbb{G}_T^4$. Denoted by $\zeta = 0$ that $Z = g_t^{\eta_1 \cdot \eta_2}$, and by $\zeta = 1$ that Z is random. In the following, we construct a simulator $\mathcal{B}$ that runs $\mathcal{A}_2$ as a subroutine to correctly guess the value of ζ.

1. Setup: $\mathcal{B}$ sends pp and $msk = y$ to $\mathcal{A}_2$.
2. Phase 1: Same as Phase 1 in the proof of Theorem 2.
3. Challenge: $\mathcal{A}_2$ selects ID_{s^*}, ID_{r^*}, and two keywords (w_0^*, w_1^*) for the challenge, with the following restrictions: (1) Neither ID_{s^*} nor ID_{r^*} has been submitted to $\mathcal{O}_{sk}$; (2) Neither $(ID_{s^*}, ID_{r^*}, w_0^*)$ nor $(ID_{s^*}, ID_{r^*}, w_1^*)$ has been submitted to $\mathcal{O}_{CLPAEKS}$. If $\neg(s^* = i^* \wedge r^* = j^*) \wedge \neg(s^* = j^* \wedge r^* = i^*)$, $\mathcal{B}$ aborts and randomly returns $\zeta' \in \{0,1\}$. Otherwise, $\mathcal{B}$ randomly selects $b \in \{0,1\}$ and sends $td^* = (td_1^*, td_2^*, td_3^*)$ to $\mathcal{A}_2$, in which

$$(\beta^*, \gamma^*) \in \mathbb{Z}_p^2, \quad k^* = H_3(ID_{s^*} \parallel ID_{r^*} \parallel Z \cdot \hat{e}(H_1(ID_{s^*}), H_1(ID_{r^*}))^y),$$

$$td_1^* = H_2(w_b^*)^{\beta^* + \frac{\gamma^*}{k^*}}, \quad td_2^* = H_2(w_b^*)^{\frac{(k^*)^3}{\beta^*} - \gamma^*}, \quad td_3^* = \frac{\beta^*}{k^*} + \frac{k^*}{\beta^*}.$$

4. Phase 2: $\mathcal{A}_2$ is allowed to access the oracles as in Phase 1, with the following restrictions:
   - Neither ID_{s^*} nor ID_{r^*} can be submitted to $\mathcal{O}_{sk}$.

- Neither $(ID_{s^*}, ID_{r^*}, w_0^*)$ nor $(ID_{s^*}, ID_{r^*}, w_1^*)$ can be submitted to $\mathcal{O}_{CLPAEKS}$.

5. Guess: $\mathcal{A}_2$ submits b'. If $b = b'$, $\mathcal{A}_2$ wins, and $\mathcal{B}$ returns $\zeta' = 0$. If $b \neq b'$, $\mathcal{A}_2$ loses, and $\mathcal{B}$ returns $\zeta' = 1$.

If $\zeta = 0$, $\mathcal{B}$ perfectly simulates Section 3.3.4, and $\mathcal{A}_2$'s probability of winning is $\epsilon + 1/2$. Otherwise, td^* is independent of w_b^*, and $\mathcal{A}_2$'s probability of winning is $1/2$. $\mathcal{B}$ aborts and randomly returns $\zeta' \in \{0, 1\}$ if its guess of the identities selected by $\mathcal{A}_2$ for challenge is wrong. Denote $\mathcal{B}$'s abortion with abt, we have

$$\Pr[\zeta' = \zeta | abt] = \frac{1}{2},$$

$$\Pr[\zeta' = \zeta | \overline{abt}] = (\epsilon + \frac{1}{2}) \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{\epsilon}{2} + \frac{1}{2},$$

$$\Pr[\overline{abt}] \geq \frac{1}{C_{q_{H_1}}^2} = \frac{2}{q_{H_1}(q_{H_1} - 1)}.$$

$\mathcal{B}$'s advantage in solving the DDH problem in $\mathbb{G}_T$ is

$$\begin{aligned}
&\mathrm{Adv}^{DDH} \\
&= \left| \Pr[\zeta' = \zeta] - \frac{1}{2} \right| \\
&= \left| \Pr[\zeta' = \zeta \wedge \overline{abt}] + \Pr[\zeta' = \zeta \wedge abt] - \frac{1}{2} \right| \\
&= \left| \Pr[\zeta' = \zeta | \overline{abt}] \cdot \Pr[\overline{abt}] + \Pr[\zeta' = \zeta | abt] \cdot \Pr[abt] - \frac{1}{2} \right| \\
&= \left| (\frac{\epsilon}{2} + \frac{1}{2}) \cdot \Pr[\overline{abt}] + \frac{1}{2} \cdot (1 - \Pr[\overline{abt}]) - \frac{1}{2} \right| \\
&= \frac{\epsilon}{2} \cdot \Pr[\overline{abt}] \\
&\geq \frac{\epsilon}{q_{H_1}(q_{H_1} - 1)}.
\end{aligned}$$

ϵ is negligible due to the intractability of the DDH problem in $\mathbb{G}_T$. This completes the proof. □

6. Performance Evaluation and Discussion

We compare our scheme with two related schemes [23,24]. The comparison includes storage overhead, computation overhead, and security. For simplicity, we only consider the following time-consuming operations:

- E: An exponentiation operation in $\mathbb{G}$.
- E_1: An exponentiation operation in $\mathbb{G}_1$.
- E_T: An exponentiation operation in $\mathbb{G}_T$.
- P: A bilinear pairing operation.
- H: A Hash-To-Point operation.

The comparison of storage overhead, computation overhead, and security is shown in Table 2, Table 3 and Table 4, respectively. Our scheme has higher storage and computation overhead. However, our scheme achieves stronger security. Besides, in practice, users may not need to encrypt all files but only a small part of files that contain sensitive information. Therefore, we consider that the storage and computation overhead paid for stronger security is affordable.

Table 2. Storage overhead comparison.

	Pakniat et al.'s [23]	Shiraly et al.'s [24]	Ours
$\lvert C\rvert$	$2\lvert\mathbb{G}_1\rvert$	$2\lvert\mathbb{G}\rvert$	$2\lvert\mathbb{G}_1\rvert + 1\lvert\mathbb{G}_T\rvert$
$\lvert td\rvert$	$1\lvert\mathbb{Z}_p\rvert$	$1\lvert\mathbb{Z}_p\rvert$	$2\lvert\mathbb{G}_1\rvert + 1\mathbb{Z}_p$

$|C|$, $|td|$: Size of the ciphertext and the trapdoor, respectively; $|\mathbb{G}|$, $|\mathbb{G}_1|$, $|\mathbb{G}_T|$, $|\mathbb{Z}_p|$: Size of an element in $\mathbb{G}$, $\mathbb{G}_1$, $\mathbb{G}_T$, and $\mathbb{Z}_p$, respectively.

Table 3. Computation overhead comparison.

	Pakniat et al.'s [23]	Shiraly et al.'s [24]	Ours
Ciphertext generation	$3E_1 + P + H$	$5E$	$2E_1 + 2E_T + 2P + 2H$
Trapdoor generation	$E_1 + P + H$	$3E$	$2E_1 + E_T + P + 2H$
Test	E_1	E	$E_T + 2P$

Table 4. Security comparison.

	Pakniat et al.'s [23]	Shiraly et al.'s [24]	Ours
CT-IND	yes	yes	yes
S-TD-IND	no	no	yes
Model	ROM	ROM	ROM
Assumption	GBDH & CDH	GDH	DBDH & DDH

CT-IND: Ciphertext indistinguishability; S-TD-IND: Strong trapdoor indistinguishability; ROM: Random oracle model.

7. Conclusions and Future Works

In this paper, we proposed an improved security model, in which a stronger version of trapdoor indistinguishability is defined. Then we proposed a new CLPAEKS scheme, which differs from the existing CLPAEKS schemes mainly in that the trapdoor is generated using two random elements in $\mathbb{Z}_p$. As far as we know, this is the first CLPAEKS scheme with provable security under the improved security model.

In the future, we will try to extend our scheme to make it support multi-receiver settings in order to cope with the scenario of group chat. Besides, considering that a file may contain multiple keywords, it would be valuable to extend our scheme to make it support multi-keyword settings. Furthermore, as quantum computing is emerging, traditional intractable problems, e.g., discrete logarithm problems, could be solved with a powerful quantum computer. Some quantum-safe cryptographic primitives were proposed (e.g., lattice-based cryptography, code-based cryptography, multivariate-based cryptography, and hash-based cryptography). Among the mentioned candidates, lattice-based cryptography is an attractive choice because it offers provable security and a good trade-off between efficiency and security [26–28]. Therefore, it is advisable to design a lattice-based CLPAEKS scheme to resist quantum computing attacks.

Author Contributions: Conceptualization, J.L., H.L., J.H. and Q.H.; methodology, J.L., H.L. and J.H.; writing—original draft preparation, J.L.; writing—review and editing, Q.H., S.M. and M.H.A.A.; supervision, Q.H., S.M. and M.H.A.A.; project administration, Q.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (No. 62272174, 61872152), Major Program of Guangdong Basic and Applied Research (No. 2019B030302008), Science and Technology Program of Guangzhou (No. 201902010081).

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PEKS	Public key encryption with keyword search
CLPAEKS	Certificateless public key authenticated encryption with keyword search
KGA	Keyword guessing attacks
DDH	Decisional Diffie–Hellman (assumption)
DBDH	Decisional Bilinear Diffie–Hellman (assumption)
GBDH	Gap Bilinear Diffie–Hellman (assumption)
CDH	Computational Diffie–Hellman (assumption)
GDH	Gap Diffie–Hellman (assumption)
IBEKS	Identity-based encryption with keyword search
PPT	Probabilistic polynomial time
KGC	Key generation center
CT-IND-CKA	Ciphertext indistinguishability under adaptive chosen-keyword attacks
S-TD-IND-CKA	Strong trapdoor indistinguishability under adaptive chosen-keyword attacks
ROM	Random oracle model

References

1. Boneh, D.; Crescenzo, G.D.; Ostrovsky, R.; Persiano, G. Public Key Encryption with Keyword Search. In Proceedings of the Advances in Cryptology—EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2–6 May 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 506–522.
2. Byun, J.W.; Rhee, H.S.; Park, H.; Lee, D.H. Off-Line Keyword Guessing Attacks on Recent Keyword Search Schemes over Encrypted Data. In Proceedings of the Secure Data Management, Third VLDB Workshop, Seoul, Korea, 10–11 September 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 75–83.
3. Yau, W.; Heng, S.; Goi, B. Off-Line Keyword Guessing Attacks on Recent Public Key Encryption with Keyword Search Schemes. In Proceedings of the Autonomic and Trusted Computing, 5th International Conference, Oslo, Norway, 23–25 June 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 100–105.
4. Rhee, H.S.; Park, J.H.; Susilo, W.; Lee, D.H. Trapdoor security in a searchable public-key encryption scheme with a designated tester. *J. Syst. Softw.* **2010**, *83*, 763–771. [CrossRef] [CrossRef]
5. Song, D.X.; Wagner, D.A.; Perrig, A. Practical Techniques for Searches on Encrypted Data. In Proceedings of the 2000 IEEE Symposium on Security and Privacy, Berkeley, CA, USA, 14–17 May 2000; IEEE Computer Society: Washington, DC, USA; Los Alamitos, CA, USA, 2000; pp. 44–55.
6. Rhee, H.S.; Park, J.H.; Lee, D.H. Generic construction of designated tester public-key encryption with keyword search. *Inf. Sci.* **2012**, *205*, 93–109. [CrossRef] [CrossRef]
7. Fang, L.; Susilo, W.; Ge, C.; Wang, J. Public key encryption with keyword search secure against keyword guessing attacks without random oracle. *Inf. Sci.* **2013**, *238*, 221–241. [CrossRef] [CrossRef]
8. Wang, C.h.; Tu, T.y. Keyword search encryption scheme resistant against keyword-guessing attack by the untrusted server. *J. Shanghai Jiaotong Univ. Sci.* **2014**, *19*, 440–442. [CrossRef] [CrossRef]
9. Huang, Q.; Li, H. An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks. *Inf. Sci.* **2017**, *403*, 1–14. [CrossRef] [CrossRef]
10. Zheng, Y. Digital Signcryption or How to Achieve Cost(Signature & Encryption) << Cost(Signature) + Cost(Encryption). In Proceedings of the Advances in Cryptology—CRYPTO 1997, 17th Annual International Cryptology Conference, Santa Barbara, CA, USA, 17–21 August 1997; Springer: Berlin/Heidelberg, Germany, 1997; pp. 165–179.
11. Noroozi, M.; Eslami, Z. Public key authenticated encryption with keyword search: Revisited. *IET Inf. Secur.* **2019**, *13*, 336–342. [CrossRef] [CrossRef]
12. Qin, B.; Chen, Y.; Huang, Q.; Liu, X.; Zheng, D. Public-key authenticated encryption with keyword search revisited: Security model and constructions. *Inf. Sci.* **2020**, *516*, 515–528. [CrossRef] [CrossRef]
13. Pan, X.; Li, F. Public-key authenticated encryption with keyword search achieving both multi-ciphertext and multi-trapdoor indistinguishability. *J. Syst. Archit.* **2021**, *115*, 102075. [CrossRef] [CrossRef]
14. Cheng, L.; Meng, F. Security analysis of Pan et al.'s "Public-key authenticated encryption with keyword search achieving both multi-ciphertext and multi-trapdoor indistinguishability". *J. Syst. Archit.* **2021**, *119*, 102248. [CrossRef] [CrossRef]
15. 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. In Proceedings of the Advances in Cryptology—CRYPTO 2005, 25th Annual International Cryptology Conference, Santa Barbara, CA, USA, 14–18 August 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 205–222.
16. Boneh, D.; Franklin, M.K. Identity-Based Encryption from the Weil Pairing. In Proceedings of the Advances in Cryptology—CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, CA, USA, 19–23 August 2001; Springer: Berlin/Heidelberg, Germany, 2001; pp. 213–229.

17. Li, H.; Huang, Q.; Shen, J.; Yang, G.; Susilo, W. Designated-server identity-based authenticated encryption with keyword search for encrypted emails. *Inf. Sci.* **2019**, *481*, 330–343. [CrossRef] [CrossRef]
18. Yanguo, P.; Jiangtao, C.; Changgen, P.; Zuobin, Y. Certificateless public key encryption with keyword search. *China Commun.* **2014**, *11*, 100–113. [CrossRef] [CrossRef]
19. Al-Riyami, S.S.; Paterson, K.G. Certificateless Public Key Cryptography. In Proceedings of the Advances in Cryptology—ASIACRYPT 2003, 9th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, 30 November–4 December 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 452–473.
20. He, D.; Ma, M.; Zeadally, S.; Kumar, N.; Liang, K. Certificateless Public Key Authenticated Encryption With Keyword Search for Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3618–3627. [CrossRef] [CrossRef]
21. Wu, L.; Zhang, Y.; Ma, M.; Kumar, N.; He, D. Certificateless searchable public key authenticated encryption with designated tester for cloud-assisted medical Internet of Things. *Ann. Telecommun.* **2019**, *74*, 423–434. [CrossRef] [CrossRef]
22. Liu, X.; Li, H.; Yang, G.; Susilo, W.; Tonien, J.; Huang, Q. Towards Enhanced Security for Certificateless Public-Key Authenticated Encryption with Keyword Search. In Proceedings of the Provable Security—13th International Conference, Cairns, Australia, 1–4 October 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 113–129.
23. Pakniat, N.; Shiraly, D.; Eslami, Z. Certificateless authenticated encryption with keyword search: Enhanced security model and a concrete construction for industrial IoT. *J. Inf. Secur. Appl.* **2020**, *53*, 102525. [CrossRef] [CrossRef]
24. Shiraly, D.; Pakniat, N.; Noroozi, M.; Eslami, Z. Pairing-free certificateless authenticated encryption with keyword search. *J. Syst. Archit.* **2022**, *124*, 102390. [CrossRef] [CrossRef]
25. Icart, T. How to Hash into Elliptic Curves. In Proceedings of the Advances in Cryptology—CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, 16–20 August 2009; Springer: Berlin/Heidelberg, Germany, 2009; pp. 303–316.
26. Ni, Z.; Kundi, D.; O'Neill, M.; Liu, W. A High-Performance SIKE Hardware Accelerator. *IEEE Trans. Very Large Scale Integr. Syst.* **2022**, *30*, 803–815. [CrossRef] [CrossRef]
27. Bisheh-Niasar, M.; Azarderakhsh, R.; Kermani, M.M. High-Speed NTT-based Polynomial Multiplication Accelerator for CRYSTALS-Kyber Post-Quantum Cryptography. *IACR Cryptol. EPrint Arch.* **2021**, *2021*, 563.
28. Tian, J.; Wu, B.; Wang, Z. High-Speed FPGA Implementation of SIKE Based on an Ultra-Low-Latency Modular Multiplier. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2021**, *68*, 3719–3731. [CrossRef] [CrossRef]