Soft Computing (2018) 22:7685–7696
https://doi.org/10.1007/s00500-018-3224-8

FOCUS

An efficient and secure searchable public key encryption scheme with privacy protection for cloud storage

Libing Wu[1] · Biwen Chen[1] · Sherali Zeadally[2] · Debiao He[3]

Published online: 14 May 2018

Abstract
In the area of searchable encryption, the searchable public key encryption (SPE) is an attractive technique in secure cloud storage. SPE assures the data confidentiality without affecting the usage of the data stored in the cloud. Furthermore, compared with the symmetric searchable encryption, SPE does not require key distribution and management. We investigate the security of the searchable public key encryption based on the traditional Boneh's framework. Although existing SPE schemes can enable users to search over encrypted data, most of these schemes are vulnerable to the file-injection attack and the insider keyword guessing attack. To mitigate these attacks, we propose an efficient and secure searchable public key encryption with privacy protection (SPE-PP). We then provide a concrete construction of SPE-PP that uses the Diffie–Hellman shared secret key, and we prove it can resist these attacks. Both the theoretical analysis and the experimental results show that our scheme achieves strong security along with high efficiency.

Keywords Privacy · Insider keyword guessing attack · File-injection attack · Searchable public key encryption · Secure cloud storage

1 Introduction

Cloud storage is addressing the needs of the increasing demand for storage and reduces the burden of maintaining large amounts of data. Since the cloud storage server may be untrusted and the outsourced data may contain sensitive information (*e.g.,* medical records, company finance data, and so on), end users usually prefer to encrypt their data (Ibtihal and Hassan 2017) before uploading to the cloud server (Hossain et al. 2017; AlZain et al. 2015; Li et al. 2017a) which may be malicious. This may make the utilization of the data more difficult. To achieve a trade-off between the data confidentiality and the data utilization, the searchable encryption (SE) becomes a promising solution. SE is an efficient mechanism wherein the data user is allowed to search over the encrypted data (Zhang et al. 2017; Liu et al. 2018).

Since Song et al. (2000) proposed searchable symmetric encryption (SSE), several efforts (Li et al. 2017c) have followed this research line and have improved the original work in terms of efficiency and security. To avoid the need of secret key management (Li et al. 2014a) and distribution, Boneh et al. (2004) proposed a public key searchable encryption (SPE) based on the asymmetric crypto-system, namely public key encryption with keyword search (PEKS). In most existing SPE schemes, the keyword space is assumed to have a low min-entropy. However, in real applications this assumption may not hold and, as a result, the insider adversary may guess the keyword corresponding to a trapdoor by launching the keyword guessing attack (KGA) Yau et al. (2008). To solve this problem, we need security schemes against insider KGA to protect the trapdoor privacy in the searchable encryption applications.

A few works (Rhee et al. 2010; Xu et al. 2013; Wang and Ty 2014; Li et al. 2015e; Liu et al. 2016; Chen et al. 2016) have proposed schemes that can mitigate the insider KGA. In these schemes, a malicious cloud server cannot get the information of the keywords by launching the keyword guessing attack. Nevertheless, their methods [which

Communicated by B. B. Gupta.

✉ Debiao He
hedebiao@163.com

1 Computer Science, Wuhan University, Wuhan, China

2 The College of Communication and Information, University of Kentucky, Lexington, USA

3 Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, China

use fuzzy trapdoor (Xu et al. 2013) or multi-server (Chen et al. 2016)] introduce other problems (*e.g.,* such as low efficiency and poor security) and are usually dependent on some strong assumptions such as the scheme proposed in Chen et al. (2016) which requires two cloud servers not to collude, which is hard to achieve in practice. The construction of a security scheme against insider KGA based on the traditional PEKS framework remains unclear.

In addition, Zhang et al. (2016) demonstrated a powerful attack against SE, namely the file-injection attack (FIA). The attack can reveal all client's queries by utilizing very few injected files and thus breaks the trapdoor privacy. For an FIA adversary, he/she first generates some injected files which contain exactly half of the number of keywords from the keyword space. Then, the adversary confirms if the trapdoor sent by the data user matches that injected files. It is worth noting that the adversary can complete the above judgment by simply observing the returned files. Finally, the adversary can learn some information about the keyword corresponding to the trapdoor. For most SPE schemes, it is easy to generate a legal ciphertext of the keyword by utilizing the public information. The FIA adversary can finish the FIA with the help of the cloud server which does not know that.

Recently, researchers have been focusing on achieving versatile SPE schemes (*e.g.,* such as the verifiable PESK Zhang et al. (2016a), the searchable attributed-based encryption Miao et al. (2017), certificateless searchable public key encryption Ma et al. (2017). However, to the best of our knowledge, most of them cannot mitigate the insider KGA and FIA at the same time. We argue that the main reason is because the legal ciphertext of a keyword can be generated by any adversary. Therefore, we propose to use the Diffie–Hellman shared secret key (DH-SSK), a natural shared key under the public key infrastructure (PKI), which can prevent the adversary from generating legal ciphertexts. It is worth noting that the work in Huang and Li (2017) introduces a searchable encryption, namely public key authenticated encryption with keyword search (PAEKS), which can resist the insider KGA, but the ciphertext of then data fails to achieve the indistinguishability against the chosen keyword attack. To protect the trapdoor privacy and data confidentiality, security designs against the insider KGA and the FIA are required to enable the wide application of SPE.

1.1 Our contributions

In this work, we propose a SPE-PP scheme which yields better efficiency and security for the privacy of data and trapdoor. We summarize our main contributions as follows:

- First, we formalize a new public key searchable encryption system called *searchable public key encryption with privacy protection* (SPE-PP) to address the security vulnerability against the file-injected attack and the insider keyword guessing attack in existing PEKS systems.
- Second, we present a concrete construction of SPE-PP scheme by leveraging DH-SSK and we perform an in-depth security analysis to show that the proposed scheme is provably secure.
- Finally, we analyze the computational complexity of the proposed SPE-PP scheme. We also implement a prototype of our scheme in a real database. The performance results obtained demonstrate the efficiency of our scheme and its suitability for deployment in practical applications.

2 Related works

Searchable encryption (SE) has been extensively investigated since it was introduced by Song et al. (2000). As an importance area of SE, searchable public key encryption (SPE) has been receiving a lot of attention (Boneh et al. 2004; Abdalla et al. 2005; Li et al. 2015a; Tomida et al. 2015; Wang et al. 2016) in recent years.

Framework From a cryptographic framework perspective, Boneh et al. (2004) first presented the framework of SPE, namely public key encryption with keyword search (PEKS), which is based on the public key infrastructure (PKI). Following this work, Abdalla et al. (2005) proposed a general method which can transform an anonymous identity-based encryption (Li et al. 2015c) to PEKS. They defined the notion of consistency in PEKS. Next, some IBEKS schemes (Tomida et al. 2015; Wang et al. 2016) have been proposed to avoid certificate management in traditional PKI. Recently, several attribute-based keyword search (ABKS) schemes(Xhafa et al. 2014; Miao et al. 2017; Sun et al. 2016; Li et al. 2017b) based on attribute-based encryption (Li et al. 2018, 2014b) have been proposed. These previous schemes not only support the multi-owner and multiuser scenario, but they also support fine-grained search authorization policy (Ye et al. 2017). In addition, Ma et al. (2017) proposed a new framework to avoid the key escrow problem which is based on the certificateless public key cryptography (CLPKC).

Security From a security perspective, the semantic security of searchable index defined by Boneh for PEKS requires a ciphertext of keyword not to leak any information about the keyword, while the security of a token for PEKS was formalized by Nishioka (Nishioka 2012), namely search pattern privacy (Gupta et al. 2016) which requires the search tokens not to reveal any information on their corresponding keywords. However, if the keyword space is small and of low min-entropy, Jeong et al. (2009) showed that it is hard to achieve the trapdoor privacy (Arriaga et al. 2014). This is because an adversary can launch the off-line keyword guessing attack (KGA) (Yau et al. 2008; Byun et al. 2006) and the file-injection attack (Zhang et al. 2016).

To address the aforementioned drawbacks, several solutions (Xu et al. 2013; Li et al. 2015b, d; Chen et al. 2015; Gupta and Badve 2017) have been proposed in the literature over the last few years. Xu et al. (2013) proposed a scheme with two trapdoors (i.e., a fuzzy trapdoor and an exact trapdoor) and claimed that their scheme can resist the insider KGA. In their scheme, the adversary only obtains the fuzzy trapdoor and therefore cannot extract the exact information about the keyword corresponding to the trapdoor. But their scheme suffers from some limitations such as security and efficiency. In Shao and Yang (2015), the authors proposed a general method to mitigate insider KGA by using the certificate authority of PKI and a deterministic digital signature. Their scheme can resist the insider KGA, but it is vulnerable to the FIA. Chen et al. (2015) introduced a new framework to protect against the insider KGA, but their scheme uses two servers and requires them not to collude. However, anyone can generate a legal trapdoor of a keyword in their scheme which will affect the privacy of data. In Huang and Li (2017), the authors defined a public key authenticated encryption with keyword search (PAEKS). Their main idea is that the ciphertext generation needs the data owner's secret key. Although their scheme can resist both the insider KGA and the FIA, the scheme fails to achieve the ciphertext indistinguishability against the chosen keyword attack.

The rest of the paper is organized as follows. We present all the preliminaries in Sect. 3. We give a concrete construction of SPE-PP and prove its the security in Sect. 4. In Sect. 5, we analyze the computational complexity and security of our scheme. Finally, we conclude the paper in Sect. 6.

3 Preliminaries

3.1 Notation

Table 1 presents the notations used throughout this paper.

3.2 Bilinear pairing

Suppose that $\mathbb{G}_1, \mathbb{G}_T$ are two cyclic groups, which have the same prime q. And $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$ is a computable bilinear map. The map e has the following features:

- *Computability* Given P, Q, we can compute $e(P, Q)$.
- *Non-degeneracy* For a generator $P \in \mathbb{G}_1$, $e(P, P)$ is a generator of $\mathbb{G}_T$.
- *Bilinearity*: Given $a, b \in \mathbb{Z}_q^*$ and $P_1, P_2 \in \mathbb{G}_1$, the equation $e(aP_1, bP_2) = e(P_1, P_2)^{ab}$ holds.

Table 1 Notations

Notation	Description
λ	System security parameter
W	Set of keywords
$\|W\|$	Size of the set W
$\mathbb{G}_1$,q	Elliptic curve group $\mathbb{G}_1$ with prime order q
P	A generator of $\mathbb{G}_1$
e	A pairing map
DH−SSK	The Diffie–Hellman shared secret key
h_1, h_2	Collision-resistant hash function
CA	Certificate authority
DU	Data user
DS	Data sender
CSP	The cloud server provider
(PK_u, sk_u)	DU's public/secret key pair
(PK_s, sk_s)	DS's public/secret key pair
$y \overset{R}{\leftarrow} \mathcal{Y}$	Random element y from the set $\mathcal{Y}$
C_{w_i}	Ciphertext of $w_i \in W$
T_{w_i}	Trapdoor of $w_i \in W$

3.3 Assumptions

Computational Diffie–Hellman assumption (CDH): Let $\mathbb{G}_1$ be a cyclic group, which has a prime order q, and P_1 be a generator of $\mathbb{G}_1$. Given the tuple $(P_1, aP_1, bP_1) \in \mathbb{G}_1$, where $a, b \overset{R}{\leftarrow} \mathbb{Z}_q^*$, there is no probabilistic polynomial time (PPT) algorithm to get $abP_1 \in \mathbb{G}_1$. We define the advantage $Adv_{\mathcal{A}}^{CDH}(\lambda)$ of an adversary $\mathcal{A}$ as $Pr[\mathcal{A}(P_1, aP_1, bP_1) \rightarrow abP_1]$. We know that the advantage $Adv_{\mathcal{A}}^{CDH}(\lambda)$ is negligible if the CDH assumption holds, where λ is a security parameter.

Decisional bilinear Diffie–Hellman assumption (DBDH). Let $\mathbb{G}_1, \mathbb{G}_T$ be cyclic groups of prime order q, P be the generator of $\mathbb{G}_1$, and $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$ be a bilinear pairing map. We define the advantage $Adv_{\mathcal{A}}^{DBDH}(\lambda)$ of a PPT adversary $\mathcal{A}$ as

$$\begin{aligned} Adv_{\mathcal{A}}^{DBDH}(\lambda) =& |Pr[\mathcal{A}(P, xP, yP, zP, e(P, P)^{xyz})] \\ & - Pr[\mathcal{A}(P, xP, yP, zP, T)]| \end{aligned}$$

where $x, y, z \overset{R}{\leftarrow} \mathbb{Z}_q^*$ and $T \overset{R}{\leftarrow} \mathbb{G}_T$. $Adv_{\mathcal{A}}^{DBDH}(\lambda)$ is negligible if the DBDH assumption holds, where λ is a security parameter.

3.4 DH shared secret key

Diffie–Hellman shared secret key (DH-SSK): Let the users **Alice** and **Bob** have the key pairs $(pk_A = aP, sk_a = a), (pk_B = bP, sk_B = b)$, respectively, where $P \in \mathbb{G}_1$ is a generator of the cyclic group $\mathbb{G}_1$ with the order q

and $a, b \in \mathbb{Z}_q^*$. **Alice** and **Bob** have a shared secret key $K = abP = sk_A pk_B = sk_B pk_A$, which is derived from the key agreement protocol (Diffie and Hellman 1976) proposed by Diffie and Hellman.

3.5 Attacks

For the SPE schemes based on the traditional PEKS framework, most of them are vulnerable to the insider KGA and the FIA.

Insider keyword guessing attack (Insider KGA). Suppose the cloud server provider (CSP) is an honest but curious adversary. We assume that the keyword space has a polynomial size. When receiving a trapdoor T_{w_i} from the data user (DU), the CSP tries to recover the keyword w_i corresponding to T_{w_i}.

The adversary first encrypts a possible keyword w_i^* to generate a ciphertext $C_{w_i^*}$. Then, it runs the test algorithm of the existing schemes based on the traditional framework of PEKS (i.e., Boneh et al. 2004; Rhee et al. 2010; Huang and Li 2017), and the inputs of algorithm are $C_{w_i^*}$ and T_{w_i}. If the test algorithm returns 1, this result illustrates that the adversary's guess is correct. Otherwise, the adversary continues to repeat the process until it finds the correct result.

File-injection attack (FIA) (Zhang et al. 2016). The attack consists of two phases, including the $GenFile$ phase and the $Guess$ phase. The $GenFile$ phase is in charge of generating the files to be injected. The $Guess$ phase is responsible for guessing the keyword w_i corresponding to the trapdoor T_{w_i}. The attack works as follows:

$GenFile$ phase :

- Identifies the keyword space W with $\{0, |W|-1\}$ written in binary.
- Generates the injected file F_i that contains the keywords whose ith bit is 1, where $i = 1, 2, \dots, \log|W|$.

$Guess$ phase:

- takes as input the trapdoor T_w which wishes to be guessed and runs the test algorithm of SPE schemes to obtain the returned files.
- determines the keyword w corresponding to T_w based on the returned result.

Figure 1 shows an example of the FIA with $|W| = 8$. Each keyword is identified with the set $\{000, 001, \dots, 111\}$, and the files F_0, F_1, F_2 contain the keywords (100, 101, 110, 111), (010, 011, 110, 111), (001, 101, 011, 111), respectively. If only $\{F_0, F_1\}$ are returned in response to the trapdoor T_{w_i}, then we know the keyword 110 corresponds to this trapdoor.

Based on the descriptions of the above attacks, the main reason why those attacks can break the privacy of user is because the adversary can obtain the ciphertext of each keyword and the trapdoors of keywords in existing SPE schemes.

	F_0	F_1	F_2
000	*N*	*N*	*N*
001	*N*	*N*	*Y*
010	*N*	*Y*	*N*
011	*N*	*Y*	*Y*
100	*Y*	*N*	*N*
101	*Y*	*N*	*Y*
110	*Y*	*Y*	*N*
111	*Y*	*Y*	*Y*

Fig. 1 An example of FIA

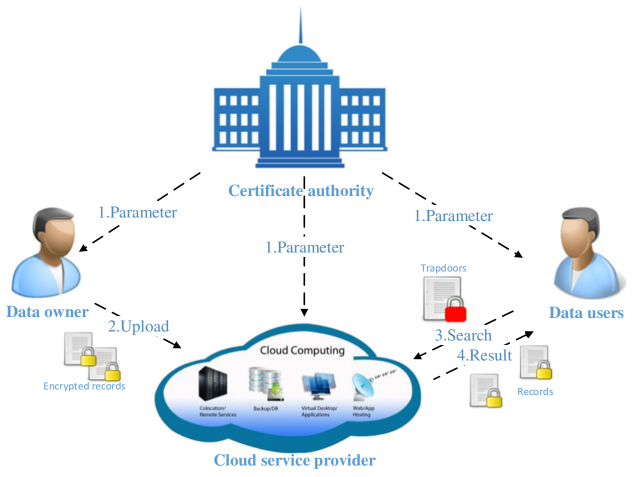

Fig. 2 System model of SPE_PP

3.6 System model

Figure 2 shows the system model of SPE_PP. The system comprises four entities which include the: certificate authority (CA), data sender (DS), data user (DU) and cloud service provider (CSP). CA selects the public parameter and publishes the entities' public key (step 1), whereas their secret keys are saved by themselves. DS generates the encrypted records by using the Diffie–Hellman shared secret key (DH-SSK) and uploads the ciphertexts to the CSP (step 2). DU is responsible for generating the trapdoor of a keyword which it wishes to be searched and sends the trapdoor to CSP (step 3). Upon receiving the trapdoor, CSP executes the test algorithm and returns the test result to DU (step 4).

Definition 1 A SPE_PP scheme consists of the following five polynomial-time algorithms:

- **Setup**(λ): It takes as input the security parameter λ and generates the public parameters $Para$.

- **KeyGen**$(Para)$: It takes as input the public parameters $Para$ and generates the two public/secret key pairs (PK_u, sk_u), (PK_s, sk_s) for DU and DS, respectively.
- **SPE_PP**$(Para, PK_u, sk_s, W)$: It takes as input the public parameters $Para$, the DU's public key PK_u, the DS's secret key sk_s and the keyword set W, and generates the ciphertext dataset C of W.
- **Trapdoor**$(Para, PK_s, sk_u, w_j)$: It takes as input the public parameters $Para$, the DS's public key PK_s, the DU's secret key sk_u and a keyword w_j, and generates the trapdoor T_{w_j} of w_j.
- **Test**$(Para, T_{w_j}, C)$: It takes as input the public parameter $Para$, the trapdoor T_{w_j} of w_j and the keyword ciphertext set C, and outputs 1 if the ciphertext C_{w_i} and T_{w_j} contain the same keyword and otherwise outputs 0.

Correctness We say that a SPE_PP scheme is correct if given $Para \leftarrow$ **Setup**(λ), $(PK_u, sk_u, PK_s, sk_s) \leftarrow$ **KeyGen**$(Para)$, $C_{w_i} \leftarrow$ **SPE_PP**$(Para, PK_u, sk_s, w_i)$, $T_{w_j} \leftarrow$ **Trapdoor**$(Para, PK_s, sk_u, w_j)$,

$$1 \leftarrow \textbf{Test}(Para, T_{w_j}, C_{w_i}) \; if\ and\ only\ if\ w_i = w_j$$

3.7 Security model

In SPE_PP, we assume that both DS and DU are honest. We also assume that CSP is an insider KGA adversary, which means that the CSP executes the SPE_PP scheme honestly and tries to obtain more information about the data and keyword. The privacy of the SPE schemes requires that the PPT adversary cannot obtain any information about the data and keywords from the ciphertexts and the trapdoors. More specifically, the adversary cannot distinguish a ciphertext/trapdoor of keyword w_0 from a ciphertext/trapdoor of keyword w_1 in the following games.

To describe the security of SPE_PP, we define the following indistinguishable games, including indistinguishability against chosen keyword attack game (IND-CKA) and indistinguishability against keyword guess attack game (IND-KGA).

The IND-CKA game shows that an adversary cannot obtain any information from the ciphertext of keyword. Simultaneously, the IND-KGA game illustrates that an adversary could not obtain any information from the trapdoor of keyword. It is worth noting that the adversary is a probabilistic polynomial-time (PPT) adversary.

IND-CKA game

The IND-CKA game consists of a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$, where the challenger $\mathcal{C}$ is responsible for establishing the system and for answering adversary inquiries. We define the following game.

Setup. Given a security parameter λ, the challenger $\mathcal{C}$ executes the **Setup**(λ) algorithm and the **KeyGen**$(Para)$ algorithm generates the public parameters $Para$ and the public/secret key pairs (PK_u, sk_u), (PK_s, sk_s) of DU and DS, respectively. Then, the challenger $\mathcal{C}$ sends $Para$, PK_u, PK_s to $\mathcal{A}$.

Phase 1. the adversary $\mathcal{A}$ takes as input $(Para, PK_u, PK_s)$ and adaptively issues queries in polynomial times as follows:

- Ciphertext Oracle $\mathcal{O}_C$: Given a keyword w, the oracle $\mathcal{O}_C$ executes the algorithm **SPE_PP** to generate the ciphertext $C_w =$ **SPE_PP**$(Para, PK_u, sk_s, w)$, and returns C_w to $\mathcal{A}$.
- Trapdoor Oracle $\mathcal{O}_T$: Given a keyword w, the oracle $\mathcal{O}_T$ executes the algorithm **Trapdoor** to generate the trapdoor $T_w =$ **Trapdoor**$(Para, PK_s, sk_u, w)$, and returns T_w to $\mathcal{A}$.

Challenge. The adversary $\mathcal{A}$ selects two keywords $(w_0, w_1) \in W$ as the challenged keywords. The only restriction is that the keyword pair has not been queried for trapdoor T_{w_0} or T_{w_1}. Upon receiving (w_0, w_1), the challenger randomly selects a bit $b \in \{0, 1\}$ and returns the ciphertext $C_{w_b} =$ **SPE_PP**$(Para, PK_u, sk_s, w_b)$ to $\mathcal{A}$.

Phase 2. The adversary $\mathcal{A}$ continues to ask for queries to $\mathcal{O}_C$ and $\mathcal{O}_T$ as in phase 1. The only restriction is that neither w_0 nor w_1 could be queried to the trapdoor oracle $\mathcal{O}_T$.

Guess. The adversary $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ and wins the game if $b' = b$.

Definition 2 A SPE_PP scheme achieves IND-CKA security if the adversary has a negligible advantage to win the above game in polynomial time, where $\mathcal{A}$'s advantage is defined as

$$Adv_{\mathcal{A}}^{CKA}(\lambda) = |Pr[b' = b] - \frac{1}{2}|$$

IND-KGA game

The IND-KGA game is similar to the IND-CKA game. The adversary $\mathcal{A}$ wants to distinguish the trapdoor T_{w_0} of w_0 from the trapdoor T_{w_1} of w_1. The game consists of a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. We define the following game.

Setup: The challenger $\mathcal{C}$ establishes the system and generates $(Para, PK_u, sk_u, PK_s, sk_s)$ as in the IND-CKA game. Then, $\mathcal{C}$ sends $(Para, PK_u, PK_s)$ to $\mathcal{A}$.

Phase 1. The adversary $\mathcal{A}$ adaptively issues queries to oracles $\mathcal{O}_C$ and $\mathcal{O}_T$ as in the IND-CKA game.

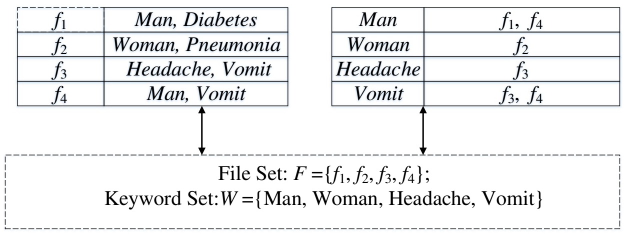

Fig. 3 An example of an inverted index-based structure

Challenge. The adversary $\mathcal{A}$ selects two keywords $(w_0, w_1) \in W$ as the challenged keywords. The only restriction is that the keyword pair has not been queried for ciphertext C_{w_0} or C_{w_1}. Upon receiving (w_0, w_1), the challenger randomly selects a bit $b \in \{0, 1\}$ and returns $T_{w_b} = \textbf{Trapdoor}(Para, PK_s, sk_u, w_b)$ to $\mathcal{A}$.

Phase 2. The adversary $\mathcal{A}$ continues to send queries to $\mathcal{O}_C$ and $\mathcal{O}_T$ as phase 1. The only restriction is that neither w_0 nor w_1 could be queried to the ciphertext oracle $\mathcal{O}_C$.

Guess. The adversary $\mathcal{A}$ outputs a guess $b^{'} \in \{0, 1\}$ and wins the game if $b^{'} = b$.

Definition 3 A SPE_PP scheme achieves IND-KGA security if the adversary has only a negligible advantage to win the above game in polynomial time, where the adversary's advantage in IND-KGA game is defined as

$$Adv_{\mathcal{A}}^{KGA}(\lambda) = |Pr[b^{'} = b] - \frac{1}{2}|$$

4 Construction

In this section, we present a specific construction of SPE_PP. Then, we analyze the correctness of our proposed scheme. Finally, we prove the security of our proposed scheme.

Our construction is more suitable for data which has the inverted index-based structure (Bost 2016), as shown in Fig. 3. We do not consider how to encrypt files, which is beyond the scope of our discussion. We assume that the DS has extracted the keyword set W from the files.

4.1 Proposed scheme

Our scheme makes use of five algorithms: the **Setup** algorithm, the **Keygen** algorithm, the **SPE_PP** algorithm, the **Trapdoor** algorithm and the **Test** algorithm. The scheme works as follows:

Setup(1^λ): Given a security parameter λ, the algorithm first selects two cyclic groups $\mathbb{G}_1, \mathbb{G}_T$ with the same prime order q and a bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$. Then, the algorithm chooses a generator P of $\mathbb{G}_1$ and two hash functions $h_1 : \mathbb{G}_1 \rightarrow \mathbb{Z}_q^*$, $h_2 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$. Finally, the algorithm publishes the public parameter $Para = (\mathbb{G}_1, \mathbb{G}_T, q, P, e, h_1, h_2)$.

KeyGen$(Para)$: DU and DS execute this algorithm to generate the public/secret key pairs. $PK_u = aP$, $PK_s = bP$ are the public keys of them and the secret keys corresponding to the public keys are $sk_u = a$, $sk_s = b$, where $a, b \in \mathbb{Z}_q^*$ are randomly selected by DU and DS, respectively. Finally, they send the public keys PK_u, PK_s to CA.

SPE_PP$(Para, PK_u, sk_s, W)$. Given the keyword set $W = \{w_1, \ldots, w_n\}$, the DS executes the following operations:

- For each keyword w_i, the DS chooses a random number $r_i \xleftarrow{R} \mathbb{Z}_q^*$, where $i = 1, 2, \ldots, n$.
- The DS computes $C_{1w_i} = r_i P$ and $C_{2w_i} = r_i Q$, where $Q = kP$ and $k = h_1(sk_s PK_u) = h_1(abP) \in \mathbb{Z}_q^*$ which is derived from the DH-SSK $key = abP$.
- The DS computes $C_{3w_i} = e(PK_u, PK_s)^{r_i h_2(w_i)}$.
- The DS obtains the ciphertext $C_{w_i} = (C_{1w_i}, C_{2w_i}, C_{3w_i})$ of w_i.

Finally, the DS uploads the encrypted records $C = \{C_{w_1}, C_{w_2}, \ldots, C_{w_n}\}$ to CSP. It is worth noting that both $e(PK_u, PK_s)$ and k can be precomputed to improve efficiency.

Trapdoor$(Para, PK_s, sk_u, w_j)$: Given a keyword w_j to be searched, where $j \in [1, n]$, the DU works as follows:

- The DU chooses a random number $r_j \xleftarrow{R} \mathbb{Z}_q^*$.
- The DU computes $T_{1w_j} = r_j P$.
- The DU computes $T_{2w_j} = r_j Q + h_2(w_j) sk_u PK_s$, where $Q = kP$ and $k = h_1(sk_u PK_s) = h_1(abP)$.
- The DU obtains the trapdoor $T_{w_j} = (T_{1w_j}, T_{2w_j})$ of w_j.

We note that k can be precomputed to improve efficiency by the DU.

Test(T_{w_j}, C): Upon receiving a trapdoor $T_{w_j} = (T_{1w_j}, T_{2w_j})$, CSP executes the following for every encrypted records $C_{w_i} = (C_{1w_i}, C_{2w_i}, C_{3w_i})$, where $i = \{1, 2, \ldots, n\}$:

- computes $E_1 = e(T_{1w_j}, C_{2w_i}) = e(r_j P, r_i Q)$.
- computes $E_2 = e(C_{1w_i}, T_{2w_j}) = e(r_i P, r_j Q + h_2(w_j) sk_u PK_s) = e(r_i P, r_j Q) e(r_i P, h_2(w_j) abP)$.
- computes $E_3 = \frac{E_2}{E_1}$ and verifies whether the equation $E_3 \overset{?}{=} C_{3w_i}$ holds. If the equation holds, CSP outputs 1; otherwise, the CSP outputs 0.

Remark We present the correctness of the **Test** algorithm below:

$$E_1 = e(T_{1w_j}, C_{2w_i}) = e(r_j P, r_i Q)$$
$$E_2 = e(C_{1w_i}, T_{2w_j}) = e(r_i P, r_j Q)e(r_i P, h_2(w_j)abP)$$
$$E_3 = \frac{E_2}{E_1} = \frac{e(r_i P, r_j Q)e(r_i P, h_2(w_j)abP)}{e(r_j P, r_i Q)}$$

Therefore, if $w_i = w_j$, then $E_3 = e(r_i P, h_2(w_j)abP) = e(PK_u, PK_s)^{r_i h_2(w_i)} = C_{3w_1}$.

4.2 Security proof

In this subsection, we prove that our scheme achieves IND-CKA security and IND-KGA security. Formally, we have the following theorems.

Theorem 1 *Our SPE_PP scheme achieves IND-CKA security if the DBDH assumption holds.*

Lemma 1 *For any PPT adversary $\mathcal{A}$, the advantage $Adv_{\mathcal{A}}^{CKA}$ in the IND-CKA game is negligible.*

Proof We define a series of games as follows.

Game 0. This is the original IND-CKA game.

Setup. The challenger $\mathcal{C}$ selects a security parameter λ and then executes the **Setup**(λ), the **KeyGen**($Para$) algorithms to generate the public parameter $Para = (\mathbb{G}_1, \mathbb{G}_T, q, P, e, h_1, h_2)$ and the public/secret key pairs $(PK_u = aP, sk_u = a), (PK_s = bP, sk_s = b)$ of DU and DS, respectively. Then, the challenger $\mathcal{C}$ sends $Para, PK_u, PK_s$ to the adversary $\mathcal{A}$ and keeps k, sk_u, sk_s secret. We assume that the hash functions h_1, h_2 are secure and collision-resistant.

Phase 1. The adversary $\mathcal{A}$ adaptively issues queries to $\mathcal{O}_C$ and $\mathcal{O}_T$, and $\mathcal{C}$ is simulated as follows:

- $\mathcal{O}_C$: The challenger $\mathcal{C}$ executes the **SPE_PP** algorithm and generates the ciphertext C_{w_i} = **SPE_PP** $(Para, PK_u, sk_s, w_i)$ of w_i.
- $\mathcal{O}_T$: The challenger executes the **Trapdoor** algorithm and generates the trapdoor T_{w_j} = **Trapdoor** $(Para, PK_s, sk_u, w_j)$ of w_j.

Challenge. The adversary $\mathcal{A}$ selects two keywords (w_0, w_1) and sends them to $\mathcal{C}$. The challenger $\mathcal{C}$ first chooses a random bit $b \in \{0, 1\}$, and then encrypts the keyword w_b as follows:

- It chooses a random number $r \leftarrow \mathbb{Z}_q^*$.
- It computes $C_{1w_b} = rP$ and $C_{2w_b} = rQ$, where $Q = kP$ and $k = h_1(sk_s PK_u) = h_1(abP)$.
- It computes $C_{3w_b} = e(PK_u, PK_s)^{rh_2(w_b)}$.

The challenger $\mathcal{C}$ sets $C_{w_b} = (C_{1w_b}, C_{2w_b}, C_{3w_b})$ as the ciphertext of w_b and sends C_{w_b} to $\mathcal{A}$.

Phase 2. The adversary $\mathcal{A}$ continues to issue queries to oracles as described above. The only restriction is that neither w_0 nor w_1 could be queried to the trapdoor oracle $\mathcal{O}_T$.

Guess. The adversary $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ on b and wins the game if $b' = b$.

According to the definition of the IND-CKA game (Definition 2), the advantage that the adversary wins in **Game** 0 is

$$Adv_{\mathcal{A}}^{\textbf{Game}\,0}(\lambda) = Adv_{\mathcal{A}}^{CKA}(\lambda)$$

Game 1. Let **Game** 1 be the same game as **Game** 0, except that the challenger chooses $T \in \mathbb{G}_T$ instead of computing $T = e(PK_u, PK_s)^r$. The challenger sends the ciphertext $C_{w_b}^* = (C_{1w_b}^* = rP, C_{2w_b}^* = rQ, C_{3w_b}^* = T^{h_2(w_b)})$. According to the **Claim** 1, we have

$$|Adv_{\mathcal{A}}^{\textbf{Game}\,1}(\lambda) - Adv_{\mathcal{A}}^{\textbf{Game}\,0}(\lambda)| \le Adv_{\mathcal{A}}^{DBDH}(\lambda)$$

where $Adv_{\mathcal{A}}^{DBDH}(\lambda)$ is negligible if the DBDH assumption holds.

Claim 1 For any PPT adversary $\mathcal{A}$, if the DBDH assumption holds, then

$$|Adv_{\mathcal{A}}^{\textbf{Game}\,1}(\lambda) - Adv_{\mathcal{A}}^{\textbf{Game}\,0}(\lambda)| \le Adv_{\mathcal{A}}^{DBDH}(\lambda)$$

Proof Given a five tuple (P, xP, yP, zP, T), where $x, y, z \overset{R}{\leftarrow} \mathbb{Z}_q^*$ and $T \in G_T$. The tuple may be a DBDH tuple, which means that $T = e(P, P)^{xyz}$. Otherwise, it is a random tuple, which means that $T \overset{R}{\leftarrow} \mathbb{G}_T$. In **Game** 0, we assume that $PK_u = xP, PK_s = yP, C_{1w_b} = zP$, then $C_{3w_b} = e(xP, yP)^{zh_2(w_b)}$. Additionally, we have $PK_u = xP, PK_s = yP, C_{1w_b} = cP, C_{3w_b} = T^{h_2(w_b)}$ in **Game** 1, where $T \overset{R}{\leftarrow} \mathbb{G}_T$. We know that it is impossible to distinguish between $T = e(P, P)^{xyz}$ and $T \overset{R}{\leftarrow} \mathbb{G}_T$ for any PPT adversary $\mathcal{A}$ if the DBDH assumption holds. Therefore, we have

$$|Adv_{\mathcal{A}}^{\textbf{Game}\,1}(\lambda) - Adv_{\mathcal{A}}^{\textbf{Game}\,0}(\lambda)| \le Adv_{\mathcal{A}}^{DBDH}(\lambda)$$

Game 2. Let **Game** 2 be the same game as **Game** 1, except that the challenge chooses $C_{w_b}^{**} = (C_{1w_b}^{**}, C_{2w_b}^{**}, C_{3w_b}^{**}) \overset{R}{\leftarrow} \mathcal{G}_1$ instead of computing $C_{w_b}^* = (C_{1w_b}^* = rP, C_{2w_b}^* = rQ, C_{3w_b}^* = T^{h_2(w_b)})$. Due to the randomness of r and T, the distributions $C^{**}_{w_b}$ and $C^{*}_{w_b}$ are indistinguishable in the adversary's view. Thus, we have

$$Adv_{\mathcal{A}}^{\textbf{Game } 2}(\lambda) = Adv_{\mathcal{A}}^{\textbf{Game } 1}(\lambda)$$

We know that the adversary can only win with probability $\frac{1}{2}$ in **Game** 2 because $C^{**}_{w_b}$ is independent of b. Thus, the advantage $Adv_{\mathcal{A}}^{\textbf{Game } 2}(\lambda) = |\frac{1}{2} - \frac{1}{2}| = 0$.

Finally, according to the **Game** 0, **Game** 1, **Game** 2, we have

$$|Adv_{\mathcal{A}}^{\textbf{Game } 2}(\lambda) - Adv_{\mathcal{A}}^{CKA}(\lambda)| \leq Adv_{A}^{DBDH}(\lambda)$$

where $Adv_{\mathcal{A}}^{\textbf{Game } 2} = 0$ and $Adv_{\mathcal{A}}^{DBDH}(\lambda)$ are negligible. Therefore, the advantage $Adv_{\mathcal{A}}^{CKA}$ that the adversary $\mathcal{A}$ wins in the IND-CKA game is negligible.

Theorem 2 *Our SPE_PP scheme achieves IND-KGA security if the CDH assumption holds.*

Lemma 2 *For any PPT adversary, the advantage $Adv_{\mathcal{A}}^{KGA}(\lambda)$ that the adversary wins in IND-KGA game is negligible.*

Proof We define a series of games as follows. **Game** 0. This is the original IND-KGA game.

Setup. The challenger $\mathcal{C}$ selects a security parameter λ and then executes the **Setup**(λ) and the **KeyGen**$(Para)$ algorithms to generate the public parameter $Para = (\mathbb{G}_1, \mathbb{G}_T, q, P, e, h_1, h_2)$ and the public/secret key pairs $(PK_u = aP, sk_u = a), (PK_s = bP, sk_s = b)$. Then, the challenger $\mathcal{C}$ sends $(Para, PK_u, PK_s)$ to $\mathcal{A}$.

Phase 1. The adversary adaptively issues queries to $\mathcal{O}_C$ and $\mathcal{O}_T$, and the challenger $\mathcal{C}$ is simulated as follows.

- $\mathcal{O}_C$: The challenger $\mathcal{C}$ executes the **SPE_PP** algorithm and generates $C_{w_i} =$ **SPE_PP**$(Para, PK_u, sk_s, w_i)$.
- $\mathcal{O}_T$: The challenger $\mathcal{C}$ executes the **Trapdoor** algorithm and generates $T_{w_j} =$ **Trapdoor**$(Para, PK_s, sk_u, w_j)$.

Challenge. The adversary $\mathcal{A}$ selects two keywords $(w_0, w_1) \in W$ and sends them to the challenger. The challenger first chooses a random bit $b \in \{0, 1\}$ and then generates the trapdoor of w_b as follows.

- It chooses a random number $r \leftarrow \mathbb{Z}_q^*$.
- It computes $T_{1w_b} = rP$.
- It computes $T_{2w_b} = rQ + h_2(w_b)sk_u PK_s$, where $Q = kP$ and $k = h_1(sk_u PK_s) = h_1(abP) \in \mathbb{Z}_q^*$.
- It obtains the trapdoor $T_{w_b} = (T_{1w_b}, T_{2w_b})$ of w_b.

The challenger sends T_{w_b} to $\mathcal{A}$.

Phase 2. The adversary $\mathcal{A}$ continues to issue queries to oracles as above. The only restriction is that neither w_0 nor w_1 could be queried to the trapdoor oracle $\mathcal{O}_T$.

Guess. The adversary outputs a guess $b' \in \{0, 1\}$ on b and wins the **Game** 0 if $b' = b$.

According to the definition of IND-KGA game (**Definition** 3), the advantage that the adversary wins in **Game** 0 is

$$Adv_{\mathcal{A}}^{\textbf{Game } 0}(\lambda) = Adv_{\mathcal{A}}^{KGA}(\lambda)$$

Let **Game** 1 be the same game as **Game** 0, except that the challenge chooses $Q^* \xleftarrow{R} \mathbb{G}_1$ instead of computing $Q = kP$, where $k = h_1(sk_u PK_s)$. Thus, the trapdoor of w_b is $T^*_{w_b} = (T^*_{1w_b} = rP, T^*_{2w_b} = (rQ^* + h_2(w_b)sk_u PK_s))$ in **Game** 1's challenge phase. We also know that it is impossible to compute $sk_u PK_s$ even if (P, PK_u, PK_s) are known for any PPT adversary. In this case, due to the randomness of r, the distributions $T^*_{2w_b}$ and T_{2w_b} are indistinguishable in the adversary' view. Thus, we have

$$|Adv_{\mathcal{A}}^{\textbf{Game } 1}(\lambda) - Adv_{\mathcal{A}}^{\textbf{Game } 0}(\lambda)| \leq Adv_{\mathcal{A}}^{CDH}(\lambda)$$

where $Adv_{\mathcal{A}}^{CDH}(\lambda)$ is negligible if the CDH assumption holds.

Game 2. Let **Game** 2 be the same game as **Game** 1, except that the challenge chooses $T^{**}_{w_b} = (T^{**}_{1w_b}, T^{**}_{2w_b}) \xleftarrow{R} \mathcal{G}_1$ instead of computing $T^*_{w_b} = (T^*_{1w_b} = rP, T^*_{2w_b} = (rQ^* + h_2(w_b)sk_u PK_s))$. Due to the randomness of r and Q^*, both $C^{**}_{w_b}$ and $C^*_{w_b}$ are the same distribution in the adversary's view. Thus, we have

$$Adv_{A}^{\textbf{Game } 2}(\lambda) = Adv_{A}^{\textbf{Game } 1}(\lambda)$$

We know that the adversary can only win with probability $\frac{1}{2}$ in **Game** 2 because T_{w_b} is independent of b. Thus, the advantage that the adversary wins in the **Game** 2 is $Adv_{\mathcal{A}}^{\textbf{Game } 2}(\lambda) = |\frac{1}{2} - \frac{1}{2}| = 0$.

Finally, according to the **Game** 0, **Game** 1, **Game** 2, we have

$$|Adv_{\mathcal{A}}^{\textbf{Game } 2}(\lambda) - Adv_{\mathcal{A}}^{KGA}(\lambda)| \leq Adv_{\mathcal{A}}^{CDH}(\lambda)$$

where $Adv_{\mathcal{A}}^{\textbf{Game } 2}(\lambda) = 0$ and $Adv_{\mathcal{A}}^{CDH}(\lambda)$ is negligible. Therefore, the advantage $Adv_{A}^{KGA}(\lambda)$ that the adversary wins in the IND-KGA game is negligible.

Resistance against insider KGA and FIA. As described in the Sect. 3, the main problem of SPE schemes is that the adversary can obtain the ciphertext of each keyword. In our proposed solution, only the data sender can generate the legal ciphertext of each keyword by leveraging the DH-SSK. When the adversary executes the **SPE_PP** algorithm, it cannot generate $C_{2w_i} = rQ = rh_1(sk_s PK_u)$P, where $sk_s PU_u$ is a DH-SSK between the DU and the DS. Thus, the ciphertext generated by the adversary cannot be used to test the trapdoor of the DU. Finally, the adversary cannot obtain any information about the trapdoor by running the **Test** algorithm. In addition, the injected files cannot be generated or will be illegal, and the FIA cannot be launched by the adversary. Thus, our construction can resist the insider KGA and FIA.

5 Performance

In this section, first, we theoretically compare our scheme with the existing schemes (Boneh et al. 2004; Rhee et al. 2010; Xu et al. 2013; Chen et al. 2016; Huang and Li 2017), which are mainly focused on solving the insider KGA in terms of computational complexity and security. Then, we conducted an empirical performance evaluation using the MIRACL library.

5.1 Analysis

Complexity The computational complexity is measured by the main operations, *e.g.*, the pairing operation T_p, the map-to-point hash function T_{mtp} and the group operations (T_{sm}, T_{et}), where T_{sm} denotes the scalar multiplication operation in $\mathbb{G}_1$ and T_{et} denotes the exponentiation operation in $\mathbb{G}_T$, where $T_p \approx T_{mtp} > 2T_{sm} > 2T_{et}$. Both the addition operation in $\mathbb{G}_1$ and the general hash operation are ignored in our analysis because it takes less time than the above operations. As shown in Table 2, the computation cost of ciphertext and trapdoor generation in our scheme is the lowest among all the other recently proposed schemes. It is worth pointing out that, in our scheme, some value (*e.g.*,, $e(PK_u, PK_s)$, $DH\text{-}SSK = sk_u PK_s = sk_s PK_u$, $k = h_1(DH\text{-}SSK)$) can be precomputed, so their computational costs are not included.

Security As shown in Table 3, the security of schemes is measured in terms of IND-CKA, the IND-KGA, insider KGA, FIA, SCF and authentication (AUT), where IND-CKA security denotes that the ciphertext of the keyword does not leak any information about the keyword, IND-KGA shows that the trapdoor of keyword does not reveal any information about the keyword, insider KGA means that the scheme can achieve some level of security against the insider KGA launched by the malicious server, FIA denotes the scheme can resist the file-injection attack, SCF means that the scheme does not require a secure channel to transmit the trapdoor, and AUT denotes that only the authenticated user can generate a legal trapdoor to search the ciphertext of keywords. Our scheme has the most secure features among all the other recently proposed schemes.

5.2 Experimental evaluation

To evaluate the efficiency of the various schemes, we implemented them (Boneh et al. 2004; Xu et al. 2013; Chen et al. 2016; Rhee et al. 2010; Huang and Li 2017) by using the MIRACL library. The experimental environment platform used consists of a Windows 7 system (64 bits) with an Inter(R) Core(TM) i5-4210U CPU @ 1.70GHz and 4.00-GB RAM. We chose a super-singular curve with the security of AES-80 (80: the key bit length of AES). The hash functions were instantiated using the SHA-256. In Table 4, we listed the running time of related operations performed in the proposed protocol.

Table 4 The running time of related operations

Operation	Running time (ms)
T_p	6.594
T_{mtp}	6.487
T_{sm}	2.635
T_{et}	0.851

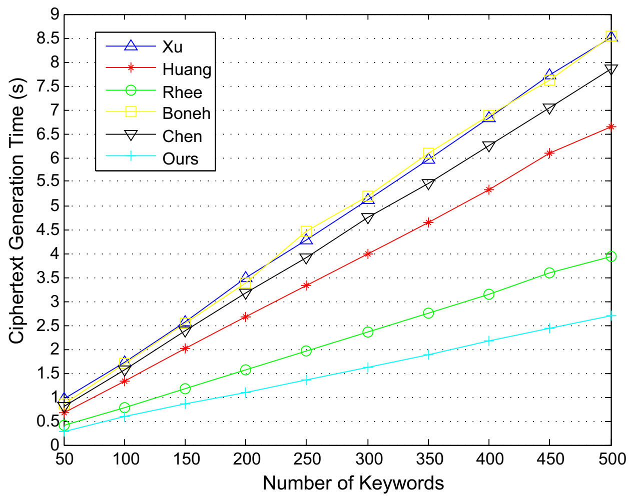

Fig. 4 Computation cost of ciphertext generation in different schemes

We compare the computational cost of our proposed scheme with the schemes in Boneh et al. (2004), Xu et al. (2013), Chen et al. (2016), Rhee et al. (2010) and Huang and Li (2017), in terms of the ciphertext generation, trapdoor generation and test algorithm. In addition, it is worth noting that the keywords were randomly selected in this experiment.

As shown in Figs. 4 and 5, in our scheme, the computation cost to generate ciphertexts and trapdoors is the lowest out of four schemes. The main reason is that our scheme does not require the map-to-point operation and avoids the pairing operation by precomputing some values.

As illustrated in Fig. 6, the computational cost of the test algorithm of our scheme is similar to that of the scheme in Huang and Li (2017). This is because that in both schemes the test algorithm requires two pairing operations. It is worth pointing out that the test algorithm is executed by the cloud server with high computing performance instead of the user, and thus, it is not a computation burden for users.

Table 2 Computation cost

Scheme	Computation cost (ms)		
	PEKS	Trapdoor	Test
Boneh et al. (2004)	$1T_p + 1T_{mtp} + 2T_{sm} \approx 18.349$	$1T_{mtp} + 1T_{sm} \approx 9.121$	$1T_p \approx 6.594$
Rhee et al. (2010)	$1T_p + 1T_{mtp} + 2T_{sm} \approx 18.349$	$3T_{sm} + 1T_{mtp} \approx 14.389$	$2T_{sm} + 1T_p \approx 11.864$
Xu et al. (2013)	$2T_{sm} + 2T_{mtp} + 2T_{et} + 2T_p \approx 33.102$	$2T_{sm} + 2T_{mtp} \approx 18.242$	$2T_p \approx 13.188$
Chen et al. (2016)	$1T_{mtp} + 4T_{sm} \approx 17.023$	$1T_{mtp} + 4T_{sm} \approx 17.023$	$7T_{sm} \approx 18.348$
Huang and Li (2017)	$1T_{mtp} + 3T_{sm} \approx 14.389$	$1T_p + 1T_{mtp} + 1T_{sm} \approx 15.715$	$2T_p \approx 13.188$
Our scheme	$2T_{sm} + 1T_{et} \approx 6.119$	$3T_{sm} \approx 7.902$	$2T_p \approx 13.188$

Table 3 Security

Scheme	Security					
	IND-CKA	IND-KGA	Insider KGA	FIA	SCF	AUT
Boneh et al. (2004)	✓	×	×	×	×	✓
Rhee et al. (2010)	✓	✓	×	×	✓	✓
Xu et al. (2013)	✓	✓	✓*(Weak)	✓	✓	✓
Chen et al. (2016)	✓	✓	✓	×	✓	×
Huang and Li (2017)	×	×	✓	✓	×	✓
Our Scheme	✓	✓	✓	✓	✓	✓

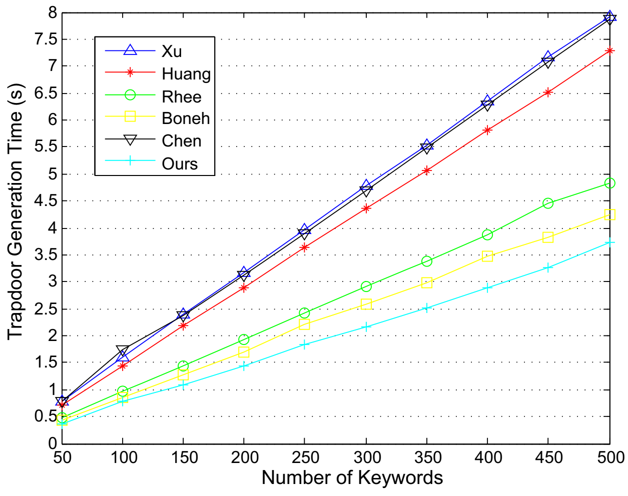

Fig. 5 Computation cost of trapdoor generation in different schemes

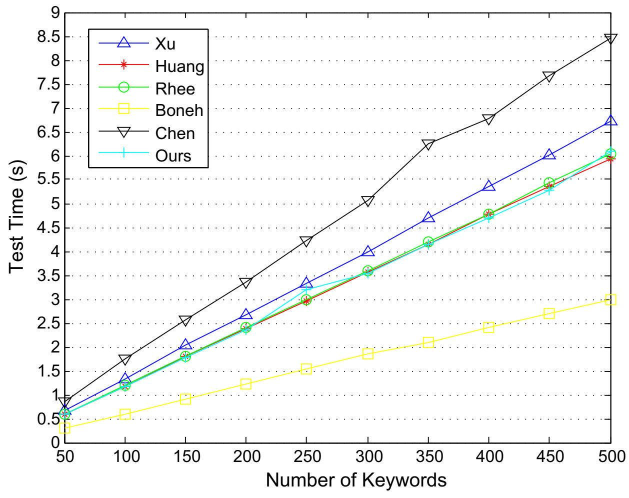

Fig. 6 Computation cost of test algorithm in different schemes

To demonstrate the practical performance of our proposed scheme, we chose the real Encron Email Dataset (Version 20150307, about 423 MB), which contains the data from about 150 users. We selected about 2000 keywords whose lengths are not less than 5 characters and the total number of occurrences is higher than 20.

As shown in Fig. 7, our scheme takes about 11.740 s to generate the ciphertexts for all keywords. It also takes about 14.629 s to generate the trapdoors corresponding to all keywords. In other words, the data user can produce 67 trapdoors in 1 s. It should be noted that the computational cost of test algorithm in Fig. 7 shows the approximate time taken by the server to finish the whole test process.

6 Conclusion

In this paper, we defined the notion of efficient and secure searchable public key encryption with privacy protection and we presented a concrete construction for SPE-PP. In our scheme, the generation of ciphertext and trapdoor requires the secret keys of the DS and the DU as input. Thus, the adversary cannot be able to launch the insider KGA and

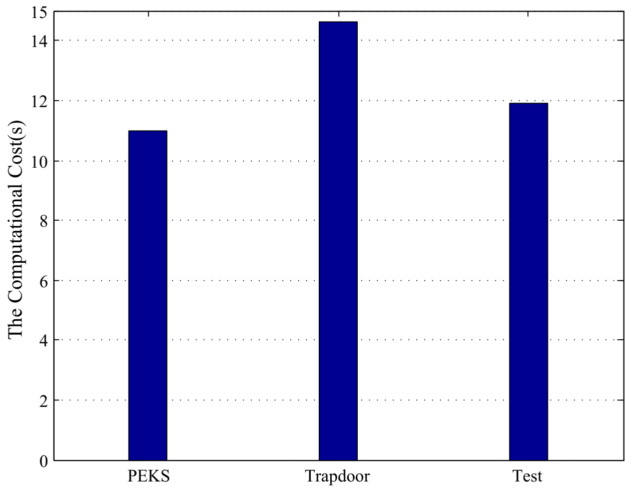

Fig. 7 Computation cost of SPK_PP in real dataset

the FIA. Then, we demonstrated that our scheme achieves the IND-CKA security and the IND-KGA security. We also analyzed performance and security of our proposed scheme against recently proposed schemes. Finally, the experimental results obtained with our scheme further demonstrate that it is practical and efficient.

Acknowledgements We thank the anonymous reviewers for their valuable comments which helped us to improve the content and presentation of this paper. The work was supported in part by the National Natural Science Foundation of China under Grants 61472287, 61772377, 61501333 and 61572379, in part by the National Key Research and Development Program of China under Grant 2017YFB0802504 and in part by the Natural Science Foundation of Hubei Province of China under Grants 2015CFA068 and 2017CFA007.

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent N/A.

References

Abdalla M, Bellare M, Catalano D, Kiltz E, Kohno T, Lange T, Malone-Lee J, Neven G, Paillier P, Shi H (2005) Searchable encryption revisited: consistency properties, relation to anonymous ibe, and extensions. Crypto, Springer 3621:205–222

AlZain MA, Li AS, Soh B, Pardede E (2015) Multi-cloud data management using shamir's secret sharing and quantum byzantine agreement schemes. Int J Cloud Appl Comput (IJCAC) 5(3):35–52

Arriaga A, Tang Q, Ryan P (2014) Trapdoor privacy in asymmetric searchable encryption schemes. In: International conference on cryptology in Africa, Springer, pp 31–50

Boneh D, Di Crescenzo G, Ostrovsky R, Persiano G (2004) Public key encryption with keyword search. Eurocrypt, Springer 3027:506–522

Bost R (2016) $o\varphi o\varsigma$: forward secure searchable encryption. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. ACM, pp 1143–1154

Byun J, Rhee H, Park HA, Lee D (2006) Off-line keyword guessing attacks on recent keyword search schemes over encrypted data. In: Secure data management, pp 75–83

Chen R, Mu Y, Yang G, Guo F, Wang X (2016) Dual-server public-key encryption with keyword search for secure cloud storage. IEEE Trans Inf Forensics Secur 11(4):789–798

Chen R, Mu Y, Yang G, Guo F, Wang X (2015) A new general framework for secure public key encryption with keyword search. In: Australasian conference on information security and privacy, Springer, pp 59–76

Diffie W, Hellman M (1976) New directions in cryptography. IEEE Trans Inf Theory 22(6):644–654

Gupta B, Badve OP (2017) Taxonomy of dos and ddos attacks and desirable defense mechanism in a cloud computing environment. Neural Comput Appl 28(12):3655–3682

Gupta B, Agrawal DP, Yamaguchi S (2016) Handbook of research on modern cryptographic solutions for computer and cyber security. In: IGI Global

Hossain MS, Muhammad G, Abdul W, Song B, Gupta B (2017) Cloud-assisted secure video transmission and sharing framework for smart cities. Future Gener Comput Syst 83:569–606

Huang Q, Li H (2017) An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks. Inf Sci 403:1–14

Ibtihal M, Hassan N et al (2017) Homomorphic encryption as a service for outsourced images in mobile cloud computing environment. Int J Cloud Appl Comput (IJCAC) 7(2):27–40

Jeong IR, Kwon JO, Hong D, Lee DH (2009) Constructing peks schemes secure against keyword guessing attacks is possible? Comput Commun 32(2):394–396

Li J, Chen X, Li M, Li J, Lee PP, Lou W (2014a) Secure deduplication with efficient and reliable convergent key management. IEEE Trans Parallel Distrib Syst 25(6):1615–1625

Li J, Huang X, Li J, Chen X, Xiang Y (2014b) Securely outsourcing attribute-based encryption with checkability. IEEE Trans Parallel Distrib Syst 25(8):2201–2210

Li H, Liu D, Dai Y, Luan TH, Shen XS (2015a) Enabling efficient multi-keyword ranked search over encrypted mobile cloud data through blind storage. IEEE Trans Emerg Top Comput 3(1):127–138

Li J, Chen X, Xhafa F, Barolli L (2015b) Secure deduplication storage systems supporting keyword search. J Comput Syst Sci 81(8):1532–1541

Li J, Li J, Chen X, Jia C, Lou W (2015c) Identity-based encryption with outsourced revocation in cloud computing. IEEE Trans Comput 64(2):425–437

Li J, Liu Z, Chen X, Xhafa F, Tan X, Wong DS (2015d) L-encdb: a lightweight framework for privacy-preserving data queries in cloud computing. Knowl Based Syst 79:18–26

Li H, Liu D, Dai Y, Luan T, Yu S (2015e) Personalized search over encrypted data with efficient and secure update in mobile clouds. IEEE Trans Emerg Topics Comput 6(1):97–109

Li T, Gupta BB, Metere R (2017a) Socially-conforming cooperative computation in cloud networks. J Parallel Distrib Comput 117:274–280

Li J, Lin X, Zhang Y, Han J (2017b) Ksf-oabe: outsourced attribute-based encryption with keyword search function for cloud storage. IEEE Trans Serv Comput 10(5):715–725

Li H, Yang Y, Dai Y, Bai J, Yu S, Xiang Y (2017c) Achieving secure and efficient dynamic searchable symmetric encryption over medical cloud data. IEEE Trans Cloud Comput. https://doi.org/10.1109/TCC.2017.2769645

Li J, Zhang Y, Chen X, Xiang Y (2018) Secure attribute-based data sharing for resource-limited users in cloud computing. Computers & Security 72(1–1):2

Liu Z, Weng J, Li J, Yang J, Fu C, Jia C (2016) Cloud-based electronic health record system supporting fuzzy keyword search. Soft Comput 20(8):3243–3255

Liu Z, Huang Y, Li J, Cheng X, Shen C (2018) Divoram: towards a practical oblivious ram with variable block size. Inf Sci. Doi: 10.1016/j.ins.2018.02.071

Ma M, He D, Kumar N, Choo KKR, Chen J (2017) Certificateless searchable public key encryption scheme for industrial internet of things. IEEE Trans Ind Inf 14(2):759–767

Miao Y, Ma J, Liu X, Li X, Jiang Q, Zhang J (2017) Attribute based keyword search over hierarchical data in cloud computing. IEEE Trans serv Comput. https://doi.org/10.1109/TSC.2017.2757467

Nishioka M (2012) Perfect keyword privacy in peks systems. ProvSec 12:175–192

Rhee HS, Park JH, Susilo W, Lee DH (2010) Trapdoor security in a searchable public-key encryption scheme with a designated tester. J Syst Softw 83(5):763–771

Shao ZY, Yang B (2015) On security against the server in designated tester public key encryption with keyword search. Inf Process Lett 115(12):957–961

Song DX, Wagner D, Perrig A (2000) Practical techniques for searches on encrypted data. In: Proceedings 2000 IEEE symposium on security and privacy, 2000. S&P 2000. IEEE, pp 44–55

Sun W, Yu S, Lou W, Hou YT, Li H (2016) Protecting your right: verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud. IEEE Trans Parallel Distrib Syst 27(4):1187–1198

Tomida K, Doi H, Mohri M, Shiraishi Y (2015) Ciphertext divided anonymous hibe and its transformation to identity-based encryption with keyword search. J Inf Process 23(5):562–569

Wang Ch, Ty Tu (2014) Keyword search encryption scheme resistant against keyword-guessing attack by the untrusted server. J Shanghai Jiaotong Univ (Sci) 19(4):440–442

Wang XF, Mu Y, Chen R, Zhang XS (2016) Secure channel free id-based searchable encryption for peer-to-peer group. J Comput Sci Technol 31(5):1012–1027

Xhafa F, Wang J, Chen X, Liu JK, Li J, Krause P (2014) An efficient phr service system supporting fuzzy keyword search and fine-grained access control. Soft Comput 18(9):1795–1802

Xu P, Jin H, Wu Q, Wang W (2013) Public-key encryption with fuzzy keyword search: a provably secure scheme under keyword guessing attack. IEEE Trans Comput 62(11):2266–2277

Yau WC, Heng SH, Goi BM (2008) Off-line keyword guessing attacks on recent public key encryption with keyword search schemes. In: Autonomic and trusted computing, pp 100–105

Ye J, Wang J, Zhao J, Shen J, Li KC (2017) Fine-grained searchable encryption in multi-user setting. Soft Comput 21(20):6201–6212

Zhang Y, Chen X, Li J, Wong DS, Li H, You I (2017) Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing. Inf Sci 379:42–61

Zhang Y, Katz J, Papamanthou C (2016) All your queries are belong to us: the power of file-injection attacks on searchable encryption. In: IACR cryptology ePrint archive 2016, pp 172

Zhang R, Xue R, Yu T, Liu L (2016a) Pvsae: a public verifiable searchable encryption service framework for outsourced encrypted data. In: 2016 IEEE international conference on web services (ICWS). IEEE, pp 428–435