sensors

MDPI

Article

Searchable Encryption Scheme for Personalized Privacy in IoT-Based Big Data

Shuai Li, Miao Li, Haitao Xu * and Xianwei Zhou

School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China; lis198707@gmail.com (S.L.); lmiao1021@gmail.com (M.L.); xwzhouli@sina.com (X.Z.)
* Correspondence: xuhaitao@ustb.edu.cn; Tel.: +86-10-6164-7796

Received: 16 January 2019; Accepted: 25 February 2019; Published: 1 March 2019

Abstract: The Internet of things (IoT) has become a significant part of our daily life. Composed of millions of intelligent devices, IoT can interconnect people with the physical world. With the development of IoT technology, the amount of data generated by sensors or devices is increasing dramatically. IoT-based big data has become a very active research area. One of the key issues in IoT-based big data is ensuring the utility of data while preserving privacy. In this paper, we deal with the protection of big data privacy in the data storage phase and propose a searchable encryption scheme satisfying personalized privacy needs. Our proposed scheme works for all file types including text, audio, image, video, etc., and meets different privacy needs of different individuals at the expense of high storage cost. We also show that our proposed scheme satisfies index indistinguishability and trapdoor indistinguishability.

Keywords: Internet of Things; big data; searchable encryption; personalized privacy needs; index indistinguishability; trapdoor indistinguishability

1. Introduction

Internet of Things (IoT) has become a significant part of our daily life over the past few years. A huge number of sensors or intelligent devices have been integrated together to interconnect people with the physical world, which also generates massive sensing data. Data generated by IoT devices are collected, disseminated, and exchanged among different people, business, and societies. With the development of IoT, the amount of data generated by organizations or individuals is increasing dramatically [1].

Although the massive data generated in the IoT environment is of significant value, exploring and using the extraordinary value of IoT data will increase the risk of privacy breach [2]. To obtain profits, the collection, storage, and reuse of our personal data poses a serious threat to our privacy. Consequently, researchers are faced with the challenge of ensuring the utility of data while preserving privacy. Various techniques have been developed to protect data privacy. Generally, these techniques for data privacy can be grouped based on the stages of big data life cycle, as follows [3].

- Data generation: In the data generation phase, access restriction, and falsifying data techniques are used.
- Data storage: The approaches in the data storage phase are mainly based on encryption techniques.
- Data processing: Anonymization techniques as well as clustering, classification, and association rule mining-based techniques are used in the data processing phase.

In this paper, we will focus on the protection of big data privacy in the data storage phase of the big data life cycle. In the IoT environment, the sensing data generated by various sensors and devices will be collected and uploaded to cloud servers, where cloud servers can provide massive

storage and cloud computing services. We know that encryption techniques are used for the protection of big data privacy in the data storage phase. When a large amount of encrypted data is stored in cloud servers, the first consideration is confidentiality of the data, which can be ensured by secure and efficient encryption schemes. However, when the data user wants to retrieve the data containing a specific keyword, the cloud server cannot respond to the data user's retrieval request, because it cannot decrypt the encrypted data. All these problems can be solved by searchable encryption schemes [4,5], such as searchable symmetric encryption [6], public key encryption with keyword search [7], etc. The searchable encryption scheme mainly includes three entities—data owner, data user, and cloud server. The data owner outsources the encrypted data to the cloud server. The data user queries the encrypted data containing a specific keyword to the cloud server. The cloud server stores and retrieves the encrypted data.

In existing searchable encryption schemes, the data user can access all the data owned by the data owner, which can result in a privacy breach for the data owner. On the one hand, the data owner may be willing to share the data with some specific data users, but not with other data users. On the other hand, the data owner may be willing to share specific data with the data user, but not willing to share other data. Therefore, the data user accesses all the data owned by the data owner, which can result in a privacy breach for the data owner. Furthermore, additional information in the data owned by the data owner can also result in a privacy breach for the data owner. Privacy is subjective, and different people have different privacy needs. For example, the hidden text in a typical Word file includes a lot of sensitive personal information [8]. However, this additional information, which may disclose the privacy of the data owner, is useless for some data users. In data mining, data preprocessing is used to transform raw data into an understandable format [9]. In natural language processing, text feature extraction is used to transform a list of words into a feature set that is usable by a classifier [10]. In speech recognition and image recognition, feature extraction is a key step [11,12]. It means that this additional information may be discarded by the data user in the feature extraction phase. In summary, the data user accessing all the data owned by the data owner will result in a privacy breach for the data owner, but will not improve the utility of the data.

In this paper, we will propose a searchable encryption scheme for personalized privacy protection in IoT-based big data. The main contributions of our proposed scheme are as follows:

- In our proposed scheme, the data owner generates the file features at different levels, and uploads the encrypted file features to the cloud server.
- The proposed scheme makes a trade-off between ensuring the utility of the data and preserving the privacy, and meets the different privacy needs of different individuals.

The rest of this paper is as follows. Section 2 discusses the recent searchable encryption scheme. Section 3 presents necessary notations and definitions. Section 4 formalizes the searchable encryption scheme for meeting the personalized privacy needs in big data and presents main security definition. Section 5 describes the detailed construction of our proposed scheme. Section 6 discusses the security of our proposed scheme. Section 7 performs real time experimental results and makes a comparison of our proposed scheme with the existing schemes. The last section is the conclusion of this paper.

2. Related Work

Several different searchable encryption schemes have been proposed to allow the data user to retrieve the encrypted data [4,5]. In this section, we give a simple review on the existing work of the searchable encryption schemes.

In 2000, Song et al. [6] first proposed a searchable encryption scheme based on the symmetric encryption algorithm, which is called searchable symmetric encryption (SSE). However, their scheme has the following limitations: it is not proven to be a secure searchable encryption scheme; the distribution of the underlying plaintexts is vulnerable to statistical attacks; and the search time is linear to the length of the document collection. To overcome these limitations, Goh et al. [13] and

Chang and Mitzenmacher [14] deployed a masked index table for SSE and introduced the notion of security for indexes. Curtmola et al. [15] generalized the security definitions of SSE and proposed two SSE schemes which are secure under the new security definitions. The search time of their schemes is linear to the number of documents. Subsequently, several SSE schemes were proposed for improvement. For example, Cash et al. [16] proposed an SSE scheme that supports conjunctive search and general Boolean queries on outsourced symmetrically encrypted data; Salam et al. [17] proposed a privacy-preserving data storage and retrieval system in cloud computing; Li et al. [18] proposed three different SSE schemes that can guard against a coercer by using the deniable encryption idea; Soleimanian et al. [19] proposed an SSE scheme to be publicly verifiable.

Although SSE schemes have high efficiency, they suffer from complicated secret key distribution. To resolve this problem, Boneh et al. [7] introduced a searchable encryption scheme based on public key cryptography, namely public key encryption with keyword search (PEKS). Waters et al. [20] showed that the PEKS schemes based on bilinear map could be applied to build encrypted and searchable auditing logs. However, the bilinear pairing operation is very complicated. Di et al. [21] introduced a PEKS scheme without bilinear pairing. The original PEKS scheme in [7] requires a secure channel to transmit the trapdoors. To overcome this limitation, Baek et al. [22] proposed a new PEKS scheme without requiring a secure channel. Byun et al. [23] introduced the off-line keyword-guessing attack (KGA) and pointed out that the original PEKS scheme in [7] was susceptible to KGA. Rhee et al. [24] proposed the notion of trapdoor indistinguishability and showed that trapdoor indistinguishability is a sufficient condition for preventing outside KGAs. Jeong et al. [25] showed that constructing secure PEKS schemes against inside KGA is impossible under the original PEKS framework in [7]. Xu et al. [26] proposed a PEKS scheme to against inside KGA. More recently, various improved PEKS schemes have been proposed. For example, Liang et al. [27] proposed a searchable attribute-based proxy re-encryption system to achieve privacy-preserving keyword search and encrypted data sharing as well as keyword update; Chen et al. [28] proposed a dual-server PEKS scheme to against inside KGA launched by the malicious server; Yang et al. [29] proposed a semantic key word searchable proxy re-encryption scheme for secure cloud storage using lattice-based cryptographic primitives; Wu et al. [30] designed an efficient and secure searchable encryption protocol using the trapdoor permutation function for cloud-based IoT; Yin et al. [31] proposed a ciphertext-policy attribute-based searchable encryption scheme to achieve keyword-based search and fine-grained access control over encrypted data.

Table 1 shows a simple comparison of some existing searchable encryption schemes. In the design of searchable encryption scheme, privacy is a key concern. However, in all the existing searchable encryption schemes, the data user can access all the data owned by the data owner, which can result in a privacy breach for the data owner.

Table 1. A comparison of some existing searchable encryption schemes.

Type	Limitation	Characteristic	Literature
SSE	need key distribution	masked index	[13,14]
		boolean queries	[16]
		against the coercer	[18]
		publicly verifiable	[19]
PEKS	lower search efficiency	without bilinear pairing	[21]
		without secure channel	[22]
		keyword update	[27]
		against inside KGA	[28]
		synonym keyword search	[29]
		fine-grained access control	[31]

3. Preliminaries

A summary of the notations used in this paper is presented in Table 2.

Table 2. Summary of notations.

Notation	Description
λ	The security parameter
$\mathbb{G}$	A cyclic group of order q
g	A generator of $\mathbb{G}$
$\mathbf{negl}(\lambda)$	A negligible function with respect to λ
$\mathbb{G}$	A cyclic group of order q
g	A generator of $\mathbb{G}$
(pk_o, sk_o)	The public/private key pairs for the data owner
(pk_u, sk_u)	The public/private key pairs for the data user
n	The number of the file of the data owner
F_i	The i-th file of the data owner($1 \leq i \leq n$)
$n_f + 1$	The number of the file feature level
l	A file feature level ($0 \leq l \leq n_f$)
L_i	The set of the authorized file feature level of F_i
F_{il}	The file feature of F_i at level l
$\mathcal{F}$	The file features set $\{F_{il} : 1 \leq i \leq n, 0 \leq l \leq n_f\}$
$\mathcal{F}'$	The encrypted file features set
W_{l_0}	The keyword set of the file features set $\{F_{il_0} : 1 \leq i \leq n\}$
w	A keyword in W_{l_0}
Ind	The index set
Ind'	The encrypted index set
$T_{w,l}$	The trapdoor with respect to w and l

The set of all binary strings of length n is denoted as $\{0,1\}^n$, and the set of all finite binary strings is denoted as $\{0,1\}^*$.

An index table (or dictionary) denotes the data structure of the form $I[key] = value$. Given a *key*, the *value* matching the *key* is returned.

A function $\mu : \mathbb{N} \to \mathbb{N}$ is negligible if for every positive polynomial $p(\cdot)$ and all sufficiently large λ, $\mu(\lambda) < \frac{1}{p(\lambda)}$. We similarly write $f(\lambda) = \mathbf{negl}(\lambda)$ to mean that there exists a negligible function $\mu(\cdot)$ such that $f(\lambda) \leq \mu(\lambda)$ for all sufficiently large λ.

The following basic cryptographic primitives can be found in [32].

A symmetric encryption scheme is a tuple $\mathcal{E} = (Gen, Enc, Dec)$ of probabilistic, polynomial-time (PPT) algorithms, where Gen takes the security parameter λ as input, and outputs a secret key k; Enc takes a key k and a message $m \in \{0,1\}^*$ as input, and outputs a ciphertext $c = Enc(k, m)$; Dec takes a key k and a ciphertext c as input, and outputs m if $c = Enc(k, m)$.

For any symmetric encryption scheme $\mathcal{E} = (Gen, Enc, Dec)$, any adversary A and any value λ for the security parameter, the chosen-plaintext attack (CPA) indistinguishability experiment $SE^{cpa}_{A,\mathcal{E}}(\lambda)$ is defined as:

1. A random key k is generated by running $Gen(\lambda)$.
2. The adversary A is given input λ and oracle access to $Enc(k, \cdot)$, and outputs a pair of messages m_0, m_1 of the same length.
3. A random bit $b \in \{0,1\}$ is chosen, and then a ciphertext $c = Enc(k, m_b)$ is computed and given to A. c is called the challenge ciphertext.
4. The adversary A continues to have oracle access to $Enc(k, \cdot)$, and outputs a bit b'.
5. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise. In the case $SE^{cpa}_{A,\mathcal{E}}(\lambda) = 1$, we say that A succeeded.

Definition 1. *A symmetric encryption scheme $\mathcal{E} = (Gen, Enc, Dec)$ is CPA-secure if for all PPT adversaries A there exists a negligible function* **negl** *such that*

$$Pr[SE^{cpa}_{A,\mathcal{E}}(\lambda) = 1] \leq \frac{1}{2} + \mathbf{negl}(\lambda),$$

where the probability is taken over the random coins used by A, as well as the random coins used in the CPA indistinguishability experiment.

For any adversary A and any value λ for the security parameter, the computational Diffie-Hellman (CDH) experiment $CDH_{A,Setup}(\lambda)$ is defined as:

1. Run $Setup(\lambda)$ to obtain output $(\mathbb{G}, q, g)$, where $\mathbb{G}$ is a cyclic group of order q (with bit length λ) and g is a generator of $\mathbb{G}$.
2. Randomly choose $a, b \in \mathbb{Z}_q$.
3. A is given $\mathbb{G}$, q, g, g^a, g^b and outputs $h \in \mathbb{G}$.
4. The output of the experiment is defined to be 1 if $h = g^{ab}$, and 0 otherwise.

Definition 2. *The CDH problem is hard relative to Setup if for all PPT adversaries A there exists a negligible function* **negl** *such that*

$$Pr[CDH_{A,Setup}(\lambda) = 1] \leq \mathbf{negl}(\lambda).$$

4. System Model

The searchable encryption scheme for personalized privacy protection mainly includes three entities, i.e., the data owner, the data user, and cloud server. The data owner outsources the encrypted file features to the cloud server. The data user queries the encrypted file features containing a specific keyword to the cloud server. The cloud server stores and retrieves the encrypted file features. As the existing searchable encryption schemes, in this paper, the data owner is considered fully trusted. The data user is considered malicious, which means it may attempt to learn more information than it can retrieve. The cloud server is considered honest but curious in the sense that it may try to learn as much information as possible from the stored encrypted data and correctly execute the searchable encryption protocol.

Given n files F_i, $1 \leq i \leq n$, and a non-negative integer l, let F_{il} denote the file feature of F_i at level l. Specially, let $F_{i0} = F_i$, i.e., the file feature of F_i at level 0 is still F_i.

Let $n_f + 1$ denote the number of the file feature level (FFL). The data owner wishes to store the file features set $\mathcal{F} = \{F_{il} : 1 \leq i \leq n, 0 \leq l \leq n_f\}$ on the cloud server. The objectives of the data owner are as follows:

- For $1 \leq i \leq n$, $0 \leq l \leq n_f$, the file feature F_{il} are stored on the cloud server such that the confidentiality of F_{il} is preserved.
- The data user queries for a keyword w and an FFL l to retrieve all authorized file features F_{il} such that $w \in F_{il_0}$ for a given l_0 in a secure and efficient way.

4.1. Formal Definition

The searchable encryption scheme for meeting the personalized privacy needs consists of the following algorithms:

- $Setup(\lambda)$: This algorithm is run by the data owner. It takes the security parameter λ as input, and outputs the global parameter Λ.
- $KeyGen(\Lambda)$: This algorithm is run by the data owner and the data user, respectively. It takes the global parameter Λ as input, and outputs public/private key pairs (pk_o, sk_o) and (pk_u, sk_u) for the data owner and the data user, respectively.
- $Store(\mathcal{F}, pk_u, sk_o)$: This algorithm is run by the data owner. It takes the file features set $\mathcal{F}$, the data user's public key pk_u and the data owner's private key sk_o as input, and outputs the encrypted file features set $\mathcal{F}'$ and the encrypted index set Ind'.
- $Trapdoor(w, l, pk_o, sk_u)$: This algorithm is run by the data user. It takes a keyword w, an FFL l, the data owner's public key pk_o, and the data user's private key sk_u as input, and outputs the trapdoor $T_{w,l}$.

- $Search(\mathcal{F}', Ind', T_{w,l})$: This algorithm is performed interactively between the cloud server and the data user. It takes the encrypted file features set $\mathcal{F}'$, the encrypted index set Ind', and the trapdoor $T_{w,l}$ as input, and outputs all authorization file features F_{il} such that $w \in F_{il_0}$ for a given l_0.

4.2. Security Definition

The searchable encryption scheme for meeting the personalized privacy needs must satisfy the index indistinguishability and the trapdoor indistinguishability under chosen keyword-FFL pair attack. As per literature [15], we define two challenge-response games $Game_I$ and $Game_T$ between the adversary A and the challenger C to show the index indistinguishability and the trapdoor indistinguishability under chosen keyword-FFL pair attack, respectively.

The adversary A plays $Game_I$ with the challenger C and attempts to distinguish an encrypted index of the given keyword-FFL pair from some encrypted indexes. If A wins $Game_I$, then A has obtained some useful information from some encrypted indexes.

$Game_I$:

Setup: Challenger C runs $Setup(\lambda)$ and $KeyGen(\Lambda)$ to generate the global parameter Λ and the public/private key pairs (pk_o, sk_o) and (pk_u, sk_u) of the data owner and the data user respectively, and sends Λ, pk_o and pk_u to A.

Adaptive query: The adversary A makes the following queries to C:

- The adversary A adaptively selects the keyword-FFL pair (w, l) for the encrypted index query. C responds with $Ind'[w']$.
- The adversary A adaptively selects the keyword-FFL pair (w, l) for the trapdoor query. C responds with $T_{w,l}$.

Challenge: The adversary A sends two challenged keyword-FFL pairs (w_0, l_0), (w_1, l_1) to C. C picks a random number $b \in \{0, 1\}$ and sends the encrypted index $Ind'[w'_b]$ of the keyword-FFL pair (w_b, l_b) to A.

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

Definition 3. *We say the searchable encryption scheme for meeting the personalized privacy needs satisfies the index indistinguishability under chosen keyword-FFL pair attack if for all PPT adversaries A there exists a negligible function* **negl** *such that*

$$Pr[A \text{ wins } Game_I] \leq \frac{1}{2} + \mathbf{negl}(\lambda).$$

Adversary A plays $Game_T$ with challenger C and attempts to distinguish a trapdoor of the given keyword-FFL pair from some trapdoors. If A wins $Game_T$, then A has obtained some useful information from some trapdoors.

$Game_T$:

Setup: C runs $Setup(\lambda)$ and $KeyGen(\lambda)$ to generate the global parameter Λ and the public/private key pairs (pk_o, sk_o) and (pk_u, sk_u) of the data owner and the data user respectively, and sends Λ, pk_o and pk_u to A.

Adaptive query: A makes the following queries to C:

- Adversary A adaptively selects the keyword-FFL pair (w, l) for the encrypted index query. C responds with $Ind'[w']$.
- Adversary A adaptively selects the keyword-FFL pair (w, l) for the trapdoor query. C responds with $T_{w,l}$.

Challenge: Adversary A sends two challenged keyword-FFL pairs (w_0, l_0), (w_1, l_1) to C. C picks a random number $b \in \{0, 1\}$ and sends the trapdoor T_{w_b, l_b} of the keyword-FFL pair (w_b, l_b) to A.

Guess: Adversary A outputs $b' \in \{0, 1\}$ and wins the game if $b' = b$.

Definition 4. *We say the searchable encryption scheme for meeting the personalized privacy needs satisfies the trapdoor indistinguishability under chosen keyword-FFL pair attack if for all PPT adversaries A there exists a negligible function* **negl** *such that*

$$Pr[A\ wins\ Game_T] \leq \frac{1}{2} + \mathbf{negl}(\lambda).$$

5. Proposed Scheme

In this section, we present our proposed searchable encryption scheme for meeting the personalized privacy needs. It consists of the following algorithms.

$Setup(\lambda)$ is run by the data owner. It takes the security parameter λ as input, and performs the following:

1. Choose a cyclic group $\mathbb{G}$ of prime order q and a generator g of $\mathbb{G}$.
2. Choose a symmetric encryption scheme $\mathcal{E} = (Gen, Enc, Dec)$.
3. Choose two collision-resistant hash functions $H_1 : \mathbb{G} \to \{0,1\}^\lambda$ and $H_2 : \{0,1\}^* \to \{0,1\}^\lambda$.
4. Set the global parameter $\Lambda = (\mathbb{G}, q, g, \mathcal{E}, H_1, H_2)$.

$KeyGen(\Lambda)$ is run by the data owner and the data user, respectively. It takes the global parameter Λ as input, and performs the following:

1. Randomly select two elements k_o and k_u in $\mathbb{Z}_q$ as the private keys of the data owner and the data user, respectively.
2. Compute g^{k_o} and g^{k_u} in $\mathbb{G}$ as the public keys of the data owner and the data user, respectively.

$Store(\mathcal{F}, pk_u, sk_o)$ is run by the data owner. It takes the file features set $\mathcal{F}$, the data user's public key $pk_u = g^{k_u}$ and the data owner's private key $sk_o = k_o$ as input, and performs the following:

1. Compute $k_1 = H_1((g^{k_u})^{k_o})$.
2. For $1 \leq i \leq n$, $0 \leq l \leq n_f$, randomly select $id_{il} \in \{0,1\}^\lambda$ as the identifier of F_{il}, run algorithm $Gen(\lambda)$ to generate the encryption key ek_{il} of F_{il}, and compute $id'_{il} = Enc(k_1, id_{il})$, $ek'_{il} = Enc(k_1, ek_{il})$, $F'_{il} = Enc(ek_{il}, F_{il})$.
3. Create the index table $\mathcal{F}'$ such that $\mathcal{F}'[id_{il}] = F'_{il}$ for every $1 \leq i \leq n$ and $0 \leq l \leq n_f$.
4. Given an FFL l_0, create the keyword set W_{l_0} of the file features set $\{F_{il_0} : 1 \leq i \leq n\}$.
5. For $w \in W_{l_0}$, compute $w' = Enc(k_1, H_2(w))$.
6. For $0 \leq l \leq n_f$, compute $l' = Enc(k_1, H_2(l))$.
7. For $1 \leq i \leq n$, construct the set L_i of the authorized FFL of the file F_i. In other words, $l \in L_i$ implies the date user has authorization to access the file feature F_{il}.
8. Create the index table Ind' such that $Ind'[w'] = \{(id'_{il}, ek'_{il}, l') : w \in F_{il_0}, l \in L_i, 1 \leq i \leq n\}$ for every $w \in W_{l_0}$.
9. Send $\mathcal{F}'$ and Ind' to the cloud server.

$Trapdoor(w, l, pk_o, sk_u)$ is run by the data user. It takes a keyword w, an FFL l, the data owner's public key $pk_o = g^{k_o}$ and the data user's private key $sk_u = k_u$ as input, and performs the following:

1. Compute $k_2 = H_1((g^{k_u})^{k_o})$.
2. Compute $T_{w,l} = Enc(k_2, H_2(w)), Enc(k_2, H_2(l))$.

$Search(\mathcal{F}', Ind', T_{w,l})$ is performed interactively between the cloud server and the data user. It takes the encrypted file features set $\mathcal{F}'$, the encrypted index set Ind' and the trapdoor $T_{w,l}$ as input, and performs the following:

1. The cloud server: Given $T_{w,l} = (T_1, T_2)$, search $Ind'[T_1]$ to obtain the set $\mathcal{S} = \{(s_1, s_2, s_3) \in Ind'[T_1] : s_3 = T_2\}$ and send $\mathcal{S}$ to the data user.
2. The data user: Given $\mathcal{S}$, create two index tables S_1 and S_2 such that $S_1[r_s] = Dec(k_2, s_1)$, $S_2[r_s] = Dec(k_2, s_2)$ for every $s = (s_1, s_2, s_3) \in \mathcal{S}$, where $k_2 = H_1((g^{k_u})^{k_o})$ and r_s ($s \in \mathcal{S}$) are randomly selected in $\{0,1\}^\lambda$. Send $\mathcal{S}_1$ to the cloud server and store $\mathcal{S}_2$.

3. The cloud server: Given $\mathcal{S}_1$, create the index table $\mathcal{R}$ such that $\mathcal{R}[r_s] = \mathcal{F}'[\mathcal{S}_1[r_s]]$ for every *key* r_s in $\mathcal{S}_1$ and send $\mathcal{R}$ to the data user.
4. The data user: Given $\mathcal{S}_2$ and $\mathcal{R}$, compute $Dec(\mathcal{S}_2[r_s], \mathcal{R}[r_s])$ for every *key* r_s in $\mathcal{S}_2$.

Remark 1. *Please note that* $k_1 = H_1((g^{k_u})^{k_o}) = H_1((g^{k_u})^{k_o}) = k_2$, *then* $T_1 = w'$, $T_2 = l'$. *Thus,* $s_1 = id'_{il}$, $s_2 = ek'_{il}$, $\mathcal{S}_1[r_s] = id_{il}$, $\mathcal{S}_2[r_s] = ek_{il}$, $\mathcal{R}[r_s] = \mathcal{F}'[\mathcal{S}_1[r_s]] = F'_{il}$ *for every* $s = (s_1, s_2, s_3) \in \mathcal{S}$, *where* $w \in F_{il_0}$, $l \in L_i$, $1 \leq i \leq n$. *Therefore, our proposed scheme is correct.*

Given an FFL l_0, creating the keyword set W_{l_0} of the file features subset $\{F_{il_0} : 1 \leq i \leq n\}$ means that F_{il_0}, $1 \leq i \leq n$ must be text. Thus, our proposed scheme works for all file types including text, audio, image, video, etc. as long as there exists an FFL l_0 such that the file feature of the file at l_0 is text.

If the authorized FFL set of the ordinal file is only created by the data owner, then the data user cannot access to the unauthorized file features, thus our proposed scheme meets the different privacy needs of different individuals.

Our proposed scheme can be extended to the multi-user scenario. Let n_o and n_u be the number of the data owners and the data users, respectively. In the multi-user scenario, the public/private key pairs are first generated for every data owner and the data user; the file features stored on the cloud server is an n_o-ary vector, where the i-th element is the encrypted file features set of the i-th data owner; the index stored on the cloud server is an $n_o \times n_u$ matrix, where the i-th row and j-th column element is the encrypted index set that the i-th data owner created for the j-th data user.

It is obvious that our proposed scheme needs increasing storage space when n_f is getting bigger. In particular, our proposed scheme has similar storage space to the existing searchable encryption schemes when $n_f = 0$.

6. Security Analysis

In this section, we show that our proposed scheme satisfies the index indistinguishability and the trapdoor indistinguishability under chosen keyword-FFL pair attack.

Theorem 1. *If* $\mathcal{E} = (Gen, Enc, Dec)$ *is CPA-Secure and the CDH problem is hard relative to* $Setup$, *then our proposed scheme satisfies the index indistinguishability under chosen keyword-FFL pair attack.*

Proof. If there exists a PPT, and adversary A wins $Game_I$, then there exists a simulator B such that $SE^{cpa}_{B,\mathcal{E}}(\lambda) = 1$ or $CDH^{cpa}_{B,Setup}(\lambda) = 1$.

In the setup phase, C runs $Setup(\lambda)$ and $KeyGen(\Lambda)$ to generate the global parameter $\Lambda = (\mathbb{G}, q, g, \mathcal{E}, H_1, H_2)$, and the public/private key pairs $(pk_o = g^{k_o}, sk_o = k_o)$ and $(pk_u = g^{k_u}, sk_u = k_u)$ of the data owner and the data user respectively. Then, C sends Λ, $pk_o = g^{k_o}$ and $pk_u = g^{k_u}$ to A.

In the adaptive query phase, assume A makes $n_q - 1$ queries to C adaptively. The q-th query can be:

- A adaptively selects the keyword-FFL pair (w_q, l_q) for the encrypted index query. C responds with $Ind'[w'_q] = \{(id'_{il_q}, ek'_{il_q}, l'_q) : w_q \in D_i, l_q \in L_i, 1 \leq i \leq n\}$, where L_i is the authorized FFL set of F_i, $id'_{il_q} = Enc(k_1, id_{il_q})$, $ek'_{il_q} = Enc(k_1, ek_{il_q})$, $l'_q = Enc(k_1, H_2(l_q))$, $k_1 = H_1((g^{k_o})^{k_u})$.
- A adaptively selects the keyword-FFL pair (w_q, l_q) for the trapdoor query. C responds with $T_{w_q,l_q} = (Enc(k_2, H_2(w_q)), Enc(k_2, H_2(l_q)))$, where $k_2 = H_1((g^{k_u})^{k_o})$.

In the challenge phase, A sends two challenged keyword-FFL pairs (w_0, l_0), (w_1, l_1) to C. C picks a random number $b \in \{0, 1\}$ and sends the encrypted index $Ind'[w_b] = \{(id'_{il_b}, ek'_{il_b}, l_b) : w_b \in D_i, l_b \in L_i, 1 \leq i \leq n\}$ of the keyword-FFL pair (w_b, l_b) to A, where $id'_{il_b} = Enc(k_1, id_{il_b})$, $ek'_{il_b} = Enc(k_1, ek_{il_b})$, $l'_b = Enc(k_1, H_2(l_b))$ and $k_2 = H_1((g^{k_1})^{k_u})$.

In the guess phase, A outputs its guess $b_1 \in \{0, 1\}$ indicating whether the challenge $Ind'[w_b]$ is the encrypted index of (w_0, l_0) or (w_1, l_1).

From the perspective of A, $id'_{il_q} = Enc(k_1, id_{il_q})$ and $ek'_{il_q} = Enc(k_1, ek_{il_q})$ are random values in $\{0,1\}^\lambda$ for every $1 \leq i \leq n$ and $2 \leq q \leq n_q$. Please note that $k_1 = H_1((g^{k_u})^{k_o}) = H_1((g^{k_o})^{k_u}) = k_2$. Then the information obtained by the adversary A in $Game_I$ was the same as the information obtained by a simulator B in the CPA indistinguishability experiment $SE^{cpa}_{A,\mathcal{E}}(\lambda)$ and in the CDH experiment $CDH_{A,Setup}(\lambda)$. Thus, if A wins $Game_I$ then $SE^{cpa}_{B,\mathcal{E}}(\lambda) = 1$ or $CDH_{B,Setup}(\lambda) = 1$, i.e.,

$$\begin{aligned} Pr[\text{A wins } Game_I] &\leq SE^{cpa}_{B,\mathcal{E}}(\lambda) + CDH_{B,setup}(\lambda) \\ &\leq \frac{1}{2} + \mathbf{negl}(\lambda). \end{aligned}$$

Therefore, our proposed scheme satisfies the index indistinguishability under chosen keyword-FFL pair attack if $\mathcal{E} = (Gen, Enc, Dec)$ is CPA-Secure and the CDH problem is hard relative to $Setup$. □

Similarly, we can prove the following theorem:

Theorem 2. *If $\mathcal{E} = (Gen, Enc, Dec)$ is CPA-Secure and the CDH problem is hard relative to Setup, then our proposed scheme satisfies the trapdoor indistinguishability under chosen keyword-FFL pair attack.*

7. Performance Analysis

As shown in Table 3, we present a comprehensive comparison of the computation cost between our proposed scheme and some existing searchable encryption schemes. The notations used in Table 3 are as follows:

1. T_{bp}: Time cost for a bilinear pairing.
2. T_h: Time cost for a hash function.
3. T_{exp}: Time cost for an exponentiation operation in $\mathbb{G}$.
4. T_{mul}: Time cost for a multiplication operation in $\mathbb{G}$.
5. T_{enc}: Time cost for an encryption process of $\mathcal{E}$.
6. T_{dec}: Time cost for a decryption process of $\mathcal{E}$.

Table 3. Computation cost: a comprehensive comparison.

Scheme	Computation		
	Storage Phase	**Trapdoor Phase**	**Search Phase**
Boneh et al. [7]	$T_{bp} + 2T_h + 2T_{exp}$	$T_h + T_{exp}$	$T_{bp} + T_h$
Rhee et al. [24]	$T_{bp} + 2T_h + 2T_{exp}$	$2T_h + 3T_{exp}$	$T_{bp} + 2T_h + 2T_{exp} + T_{mul}$
Xu et al. [26]	$2T_{bp} + 4T_h + 4T_{exp}$	$2T_h + 2T_{exp}$	$2T_{bp} + 2T_h$
Chen et al. [28]	$T_h + 4T_{exp} + 2T_{mul}$	$T_h + 4T_{exp} + 2T_{mul}$	$7T_{exp} + 3T_{mul}$
Our scheme	$T_{exp} + 3T_h + 5T_{enc}$	$T_{exp} + 3T_h + 2T_{enc}$	$T_{exp} + T_h + 2T_{dec}$

To meet the basic security level for comparison, SHA-256 and AES-256 is selected as the collision-resistant hash function and the symmetric encryption scheme, respectively. The cyclic group $\mathbb{G}$ of order q is generated by a point on an elliptic curve $E(F_p)$, where q and p are the 256-bits and 521-bits prime numbers, respectively. To evaluate the efficiency of the five schemes, we perform our experiments on a computer with 2.4 GHz Intel Core i7 and 8 GB RAM.

As shown in Figures 1–3, our proposed scheme is the most efficient in storage phase and search phase. In trapdoor phase, our proposed scheme has a higher computational cost than that of Boneh et al. [7], although it is still lower than other schemes. In summary, the performance of our proposed scheme is more efficient than four schemes studied in [7,24,26,28].

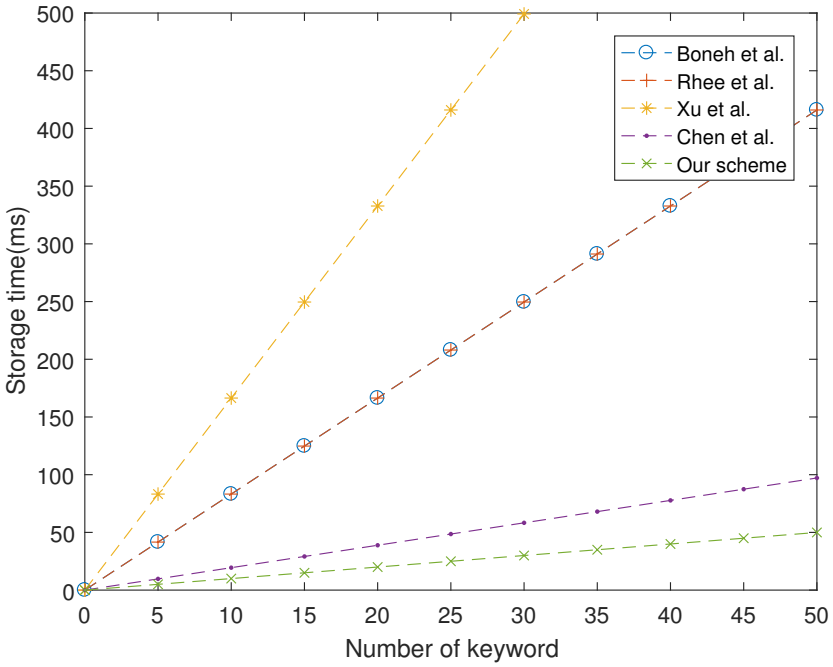

Figure 1. Computation cost at storage phase.

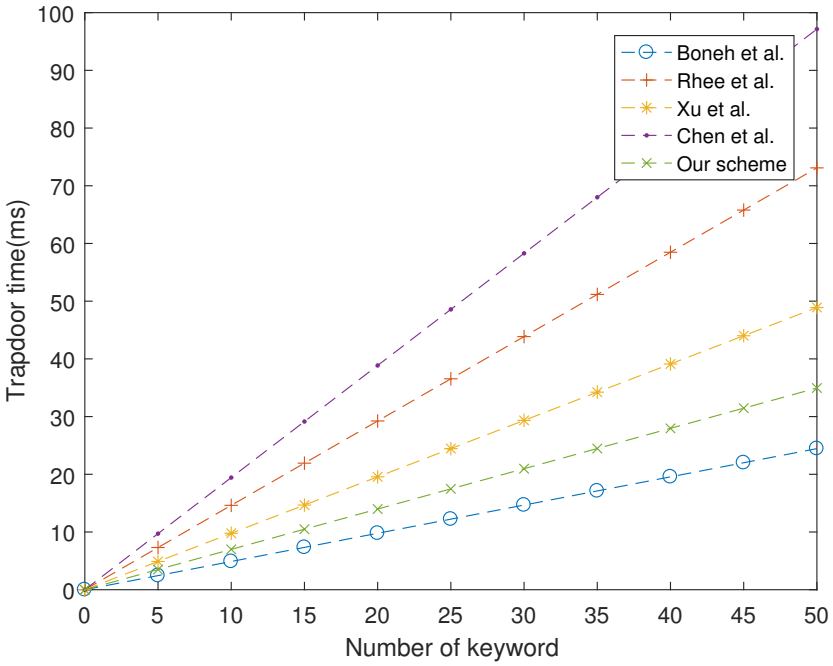

Figure 2. Computation cost at trapdoor phase.

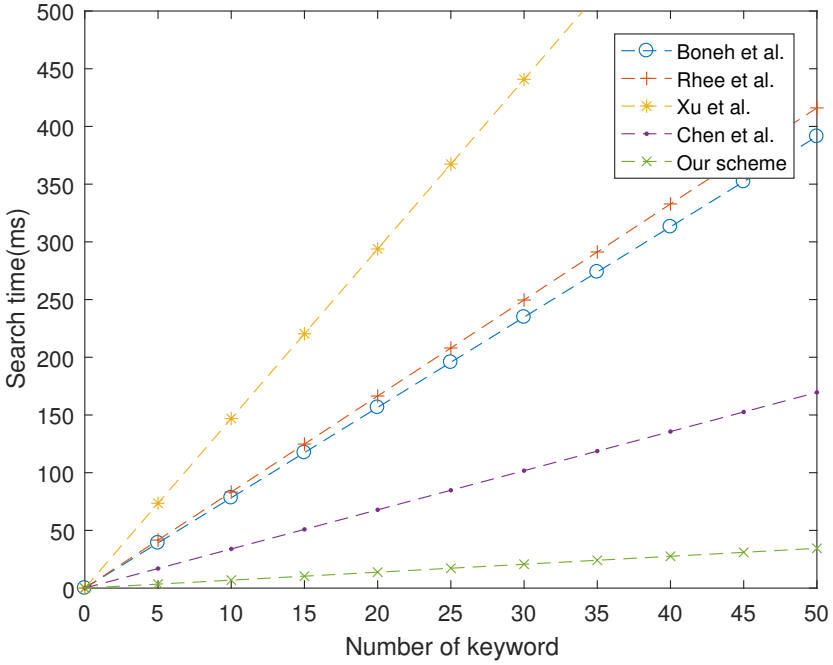

Figure 3. Computation cost at search phase.

8. Conclusions

In this paper, we have proposed a searchable encryption scheme for meeting personalized privacy needs. Our proposed scheme mainly includes three entities, i.e., the data owner, the data user, and cloud server. The data owner outsources the encrypted file features to the cloud server. The data user queries the encrypted file features containing a specific keyword to the cloud server. The cloud server stores and retrieves the encrypted file features. Compared with the existing searchable encryption schemes, our proposed scheme works for all file types including text, audio, image, video, etc., and meets different privacy needs of different individuals at the expense of high storage cost. We also show that our proposed scheme satisfies index indistinguishability and trapdoor indistinguishability under chosen keyword-FFL pair attack. In other words, our proposed scheme is secure against inside KGA. Performance analysis shows that our proposed scheme is efficient in storage phase, trapdoor phase, and search phase.

Considering the decreasing costs of storage, storage cost is not a problem if $n_f + 1$, i.e., the number of the FFL is small in our proposed scheme. However, storage cost is still a problem if n_f is too large in our proposed scheme. Thus, choosing an appropriate n_f is an important work in the future.

Author Contributions: Writing—original draft preparation, S.L.; writing—review and editing, M.L., H.X. and W.Z.

Funding: This work is supported by the National Key R&D Program of China (No. 2018YFB1003905) and the National Natural Science Foundation of China under Grant (No. U1603116, No. 61701020).

Acknowledgments: The authors would like to thank the editor and the anonymous reviewers for their valuable comments and suggestions that improved the quality of this paper.

Conflicts of Interest: The authors declare no conflicts of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

IOT	Internet of Things
SSE	searchable symmetric encryption
PKES	public key encryption with keyword search
KGA	keyword guessing attac
FFL	The file feature level

References

1. Lohr, S. The age of big data. *New York Times*, 11 February 2012.
2. John Walker, S. *Big Data: A Revolution That Will Transform How We Live, Work, and Think*; Houghton Mifflin Harcourt: Boston, MA, USA, 2014.
3. Mehmood, A.; Natgunanathan, I.; Xiang, Y.; Hua, G.; Guo, S. Protection of big data privacy. *IEEE Access* **2016**, *4*, 1821–1834. [CrossRef]
4. Bösch, C.; Hartel, P.; Jonker, W.; Peter, A. A survey of provably secure searchable encryption. *ACM Comput. Surv.* **2015**, *47*, 18. [CrossRef]
5. Poh, G.S.; Chin, J.J.; Yau, W.C.; Choo, K.K.R.; Mohamad, M.S. Searchable symmetric encryption: Designs and challenges. *ACM Comput. Surv.* **2017**, *50*, 40. [CrossRef]
6. Song, D.X.; Wagner, D.; 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; pp. 44–55.
7. Boneh, D.; Di Crescenzo, G.; Ostrovsky, R.; Persiano, G. Public key encryption with keyword search. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2–6 May 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 506–522.
8. Byers, S. Information leakage caused by hidden data in published documents. *IEEE Secur. Privacy* **2004**, *2*, 23–27. [CrossRef]
9. Hand, D.J. Principles of data mining. *Drug Safety* **2007**, *30*, 621–622. [CrossRef] [PubMed]
10. Feldman, R.; Sanger, J. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*; Cambridge University Press: Cambridge, UK, 2007.
11. Hirsch, H.G.; Pearce, D. The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In Proceedings of the ASR2000-Automatic Speech Recognition: Challenges for the New Millenium ISCA Tutorial and Research Workshop (ITRW), Beijing, China, 16–20 October 2000.
12. Hong, Z.Q. Algebraic feature extraction of image for recognition. *Pattern Recogn.* **1991**, *24*, 211–219. [CrossRef]
13. Goh, E.J. Secure indexes. *IACR Cryptol. ePrint Arch.* **2003**, *2003*, 216.
14. Chang, Y.C.; Mitzenmacher, M. Privacy preserving keyword searches on remote encrypted data. In Proceedings of the International Conference on Applied Cryptography and Network Security, New York, NY, USA, 7–10 June 2005; pp. 442–455.
15. Curtmola, R.; Garay, J.; Kamara, S.; Ostrovsky, R. Searchable symmetric encryption: Improved definitions and efficient constructions. *J. Comput. Secur.* **2011**, *19*, 895–934. [CrossRef]
16. Cash, D.; Jarecki, S.; Jutla, C.; Krawczyk, H.; Roşu, M.C.; Steiner, M. Highly-scalable searchable symmetric encryption with support for boolean queries. In *Advances in Cryptology–CRYPTO 2013*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 353–373.
17. Salam, M.I.; Yau, W.C.; Chin, J.J.; Heng, S.H.; Ling, H.C.; Phan, R.C.; Poh, G.S.; Tan, S.Y.; Yap, W.S. Implementation of searchable symmetric encryption for privacy-preserving keyword search on cloud storage. *Hum. Centr. Comput. Inf. Sci.* **2015**, *5*, 19. [CrossRef]
18. Li, H.; Zhang, F.; Fan, C.I. Deniable searchable symmetric encryption. *Inf. Sci.* **2017**, *402*, 233–243. [CrossRef]
19. Soleimanian, A.; Khazaei, S. Publicly verifiable searchable symmetric encryption based on efficient cryptographic components. *Des. Codes Cryptogr.* **2019**, *87*, 123–147. [CrossRef]
20. Waters, B.R.; Balfanz, D.; Durfee, G.; Smetters, D.K. *Building an Encrypted and Searchable Audit Log*; NDSS: San Diego, CA, USA, 2004; Volume 4, pp. 5–6.

21. Di Crescenzo, G.; Saraswat, V. Public key encryption with searchable keywords based on Jacobi symbols. In Proceedings of the International Conference on Cryptology in India, Chennai, India, 9–13 December 2007; pp. 282–296.
22. Baek, J.; Safavi-Naini, R.; Susilo, W. Public key encryption with keyword search revisited. In Proceedings of the International conference on Computational Science and Its Applications, Perugia, Italy, 30 June–3 July 2008; pp. 1249–1259.
23. Byun, J.W.; Rhee, H.S.; Park, H.A.; Lee, D.H. Off-line keyword guessing attacks on recent keyword search schemes over encrypted data. In Proceedings of the Workshop on Secure Data Management, Seoul, Korea, 10–11 September 2006; pp. 75–83.
24. 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]
25. Jeong, I.R.; Kwon, J.O.; Hong, D.; Lee, D.H. Constructing PEKS schemes secure against keyword guessing attacks is possible? *Comput. Commun.* **2009**, *32*, 394–396. [CrossRef]
26. Xu, P.; Jin, H.; Wu, Q.; Wang, W. Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack. *IEEE Trans. Comput.* **2013**, *62*, 2266–2277. [CrossRef]
27. Liang, K.; Susilo, W. Searchable attribute-based mechanism with efficient data sharing for secure cloud storage. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 1981–1992. [CrossRef]
28. Chen, R.; Mu, Y.; Yang, G.; Guo, F.; Wang, X. Dual-server public-key encryption with keyword search for secure cloud storage. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 789–798. [CrossRef]
29. Yang, Y.; Zheng, X.; Chang, V.; Tang, C. Semantic keyword searchable proxy re-encryption for postquantum secure cloud storage. *Concurr. Comput. Pract. Exp.* **2017**, *29*, e4211. [CrossRef]
30. Wu, L.; Chen, B.; Zeadally, S.; He, D. An efficient and secure searchable public key encryption scheme with privacy protection for cloud storage. *Soft Comput.* **2018**, *22*, 7685–7696. [CrossRef]
31. Yin, H.; Zhang, J.; Xiong, Y.; Ou, L.; Li, F.; Liao, S.; Li, K. CP-ABSE: A Ciphertext-Policy Attribute-Based Searchable Encryption Scheme. *IEEE Access* **2019**, *7*, 5682–5694. [CrossRef]
32. Lindell, Y.; Katz, J. *Introduction to Modern Cryptography*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2014.