Computing and Informatics, Vol. 32, 2013, 723–738

EFFICIENT MULTI-USER KEYWORD SEARCH OVER ENCRYPTED DATA IN CLOUD COMPUTING

Jin Li

Department of Computer Science, Guangzhou University
Guangzhou, P.R. China
e-mail: lijin@gzhu.edu.cn

Xiaofeng Chen

Key Laboratory of Computer Networks and Information Security
Xidian University, Xi'an, P.R. China
e-mail: xfchen@xidian.edu.cn

Communicated by Ilsun You

Abstract. As cloud computing becomes prevalent, more and more sensitive information are being centralized into the cloud. For the protection of data privacy, sensitive data usually have to be encrypted before outsourcing, which makes effective data utilization a very challenging task. In this paper, we propose a new method to enable effective fuzzy keyword search in a multi-user system over encrypted cloud data while maintaining keyword privacy. In this new system, differential searching privileges are supported, which is achieved with the technique of attribute-based encryption. Edit distance is utilized to quantify keywords similarity and develop fuzzy keyword search technique, which achieve optimized storage and representation overheads. We further propose a symbol-based trie-traverse searching scheme to improve the search efficiency. Through rigorous security analysis, we show that our proposed solution is secure and privacy-preserving, while correctly realizing the goal of fuzzy keyword search with multiple users.

Keywords: Keyword search, fuzzy, multi-user, cloud computing

Mathematics Subject Classification 2010: 94A60

1 INTRODUCTION

Cloud computing, the new term for the long dreamed vision of computing as a utility, enables convenient, on-demand network access to a centralized pool of configurable computing resources (e.g., networks, applications, and services) that can be rapidly deployed with great efficiency and minimal management overhead.

As cloud computing becomes prevalent, more and more sensitive information are being centralized into the cloud, such as emails, personal health records, private videos and photos, company finance data, government documents, etc. By storing their data into the cloud, the data owners can be relieved from the burden of data storage and maintenance so as to enjoy the on-demand high quality data storage service. However, the fact that data owners and cloud server are not in the same trusted domain may put the outsourced data at risk, as the cloud server may no longer be fully trusted in such a cloud environment due to a number of reasons [1, 2]: the cloud server may leak data information to unauthorized entities or be hacked. It follows that sensitive data usually should be encrypted prior to outsourcing for data privacy and combatting unsolicited accesses. However, data encryption makes effective data utilization a very challenging task given that there could be a large amount of outsourced data files [3, 4, 5, 6]. Moreover, in cloud computing, data owners may share their outsourced data with a large number of users owning different privileges. The individual users might want to only retrieve certain specific data files they are interested in during a given session. One of the most popular ways is to selectively retrieve files through keyword-based search instead of retrieving all the encrypted files back which is completely impractical in cloud computing scenarios. Beside this, data encryption also demands the protection of keyword privacy since keywords usually contain important information related to the data files.

Thus, keyword privacy should also be ensured so that no unauthorized entity is able to get any sensitive information from the search operations. All these problems make effective data utilization and search a challenging task, especially when there could be a large number of on-demand data users and data files.

Although allowing for performing searches securely and effectively, the existing searchable encryption techniques do not suit for cloud computing scenario since they support only *exact* keyword search. That is, there is no tolerance of minor typos and format inconsistencies which, on the other hand, are typical user searching behavior and happen very frequently. Such user searching behavior is especially inevitable in cloud computing because the data owners may share their outsourced cloud data with a large number of data users through on-demand authorization. As common practice, users may search and retrieve the data of their respective interests using any keywords they might come up with. Recently, Li et al. [7, 8, 9, 10] proposed a new way to enable fuzzy keyword search over encrypted data by introducing the edit distance in the encrypted keywords. However, they have not considered the construction in a multi-user system with differential searching privileges. Furthermore, they can only support the direct and simple search at the server side.

In this paper, we propose an efficient fuzzy keyword search scheme with multi-user over encrypted cloud data while maintaining keyword privacy and supporting authorized search. A gram-based fuzzy keyword set is presented, which is a different technique from the wildcard-based method [7]. Based on the constructed fuzzy keyword sets, we further propose an advanced symbol-based trie-traverse searching scheme, where a multi-way tree structure is built up using symbols transformed from the fuzzy keywords. Through rigorous security analysis, we show that the proposed solution is secure and privacy-preserving, while supporting multi-user searching with differential privileges.

The rest of the paper is organized as follows: Section 2 summarizes the features of related work. Section 3 introduces the system model, threat model, our design goal and briefly describes some necessary background for the techniques used in this paper. Section 4 provides detailed description of our proposed schemes. Section 5 presents the security analysis. Finally, Section 6 concludes the paper.

2 RELATED WORK

Traditional searchable encryption [11, 12, 13, 14, 15, 16, 17, 18, 19] has been widely studied in the context of cryptography. Among those works, most are focused on efficiency improvements and security definition formalizations. The first construction of searchable encryption was proposed by Song et al. [12], in which each word in the document is encrypted independently under a special two-layered encryption construction. Goh [13] proposed to use Bloom filters to construct the indexes for the data files. For each file, a Bloom filter containing trapdoors of all unique words is built up and stored on the server. To search for a word, the user generates the search request by computing the trapdoor of the word and sends it to the server. Upon receiving the request, the server tests if any Bloom filter contains the trapdoor of the query word and returns the corresponding file identifiers. To achieve more efficient search, Chang et al. [16] and Curtmola et al. [17] both proposed similar "index" approaches, where a single encrypted hash table index is built for the entire file collection. In the index table, each entry consists of the trapdoor of a keyword and an encrypted set of file identifiers whose corresponding data files contain the keyword. As a complementary approach, Boneh et al. [14] presented a public-key based searchable encryption scheme, with an analogous scenario to that of [12]. In their construction, anyone with the public key can write to the data stored on the server but only authorized users with the private key can search. As an attempt to enrich query predicates, conjunctive keyword search, subset query and range query over encrypted data have also been proposed in [18, 20]. Note that all these existing schemes support only exact keyword search, and thus are not suitable for cloud computing.

Private matching [21], as another related notion, has been studied mostly in the context of secure multiparty computation to let different parties compute some function of their own data collaboratively without revealing their data to the others.

These functions could be intersection or approximate private matching of two sets, etc. [22]. Private information retrieval [23] is an often-used technique to retrieve the matching items secretly, which has been widely applied in information retrieval from database and usually incurs unexpectedly computation complexity.

3 PROBLEM FORMULATION

3.1 System Model

In this paper, we consider a cloud data system consisting of data owner, data user and cloud server. Given a collection of n encrypted data files $\mathcal{C} = (\mathrm{F}_1, \mathrm{F}_2, \ldots, \mathrm{F}_N)$ stored in the cloud server, a predefined set of distinct keywords $W = \{w_1, w_2, \ldots, w_p\}$ and a set of searching privileges $PS = \{P_1, P_2, \cdots, P_k\}$, the cloud server provides the search service for the authorized users over the encrypted data $\mathcal{C}$. We assume the authorization between the data owner and users is appropriately done. An authorized user types in a request to selectively retrieve data files of his/her interest. The cloud server is responsible for mapping the searching request to a set of data files, where each file is indexed by a file ID and linked to a set of keywords. The fuzzy keyword search scheme returns the search results according to the following rules:

1. if the user's searching input exactly matches the pre-set keyword, the server is expected to return the files containing the keyword;
2. if there exist typos and/or format inconsistencies in the searching input, the server will return the closest possible results based on pre-specified similarity semantics.

3.2 Security Model

We consider a semi-trusted server. In this work, we just consider Honest but Curious Cloud Servers. That is to say, cloud servers will follow our proposed protocol, but try to find out as much secret information as possible based on their inputs. More specifically, we assume cloud servers are more interested in contents of data files stored. A secure communication channel between users and cloud servers is required. Users would try to access data files either within or out of the scope of their access privilege. Two kinds of attackers are considered in this system, that is,

1. external attackers, including the server, revoked users and other unauthorized users;
2. internal attackers who share their privileges with other users without such ones.

Even though data files are encrypted, the cloud server may try to derive other sensitive information from users' search requests while performing keyword-based search over $\mathcal{C}$. Thus, the search should be conducted in a secure manner that allows data files to be securely retrieved while revealing as little information as possible to the cloud server.

In this paper, when designing fuzzy keyword search scheme, we will follow the security definition deployed in the traditional searchable encryption [17]. More specifically, it is required that nothing should be leaked from the remotely stored files and index beyond the outcome and the pattern of search queries.

We address the problem of supporting efficient yet privacy-preserving fuzzy keyword search services over encrypted cloud data. Specifically, we have the following goals:

1. to explore different mechanisms for constructing storage-efficient fuzzy keyword sets;
2. to design efficient and effective fuzzy search schemes based on the constructed fuzzy keyword sets;
3. to validate the security and evaluate the performance by conducting extensive experiments.

3.3 Preliminaries

Edit Distance. There are several methods to quantitatively measure the string similarity. In this paper, we resort to the well-studied edit distance [25] for our purpose. The edit distance $\mathsf{ed}(w_1, w_2)$ between two words w_1 and w_2 is the number of operations required to transform one of them into the other. The three primitive operations are:

1. Substitution – changing one character to another in a word;
2. Deletion – deleting one character from a word;
3. Insertion – inserting a single character into a word.

Given a keyword w, we let $S_{w,d}$ denote the set of words w' satisfying $\mathsf{ed}(w, w') \leq d$ for a certain integer d.

Fuzzy Keyword Search. Using edit distance, the definition of fuzzy keyword search can be formulated as follows: Given a collection of n encrypted data files $\mathcal{C} = (F_1, F_2, \ldots, F_N)$ stored in the cloud server, a set of distinct keywords $W = \{w_1, w_2, \ldots, w_p\}$ with predefined edit distance d, and a searching input (w, k) with edit distance k $(k \leq d)$, the execution of fuzzy keyword search returns a set of file IDs whose corresponding data files possibly contain the word w, denoted as FID_w: if $w = w_i \in W$, then return FID_{w_i}; otherwise, if $w \notin W$, then return $\{FID_{w_i}\}$, where $\mathsf{ed}(w, w_i) \leq k$. Note that the above definition is based on the assumption that $k \leq d$. In fact, d can be different for distinct keywords and the system will return $\{FID_{w_i}\}$ satisfying $\mathsf{ed}(w, w_i) \leq \min\{k, d\}$ if exact match fails.

Trapdoors of Keywords. Trapdoors of the keywords can be realized by applying a one-way function f, which is similar as [13, 11, 17]: Given a keyword w_i and a secret key sk, we can compute the trapdoor of w_i as $T_{w_i} = f(sk, w_i)$.

Attribute-based encryption. Attribute-based encryption (ABE), which enables public key based one-to-many encryption, where differential yet flexible access rights can be assigned to individual users. There are two methods in cryptography to realize the fine-grained access control based on ABE: key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE) [24]. CP-ABE is different from KP-ABE in the sense that, in CP-ABE, the data owner is able to assign certain access policy for each file. When a file is being encrypted, it will be associated with an access structure over a predefined set of attributes. The user will only be able to access the underlying file if his/her attributes match the access structure specified in the ciphertext while in KP-ABE, the access policy is bounded with each user, not with each file. Thus, we explore how to utilize CP-ABE to manage the keys and files stored in cloud computing. Denote $\mathsf{ABE} = (Setup_{ABE}, KeyGen_{ABE}, Enc_{ABE}, Dec_{ABE})$ as an attribute-based encryption scheme, where $Setup_{ABE}$ is the setup algorithm with a predefined security parameter, $KeyGen_{ABE}$ is the key-issuing algorithm running by the attribute authority, Enc_{ABE} and Dec_{ABE} are the encryption and decryption algorithms, respectively.

4 CONSTRUCTIONS OF EFFECTIVE FUZZY KEYWORD SEARCH IN CLOUD

The key idea behind our secure fuzzy keyword search is two-fold:

1. building up fuzzy keyword sets that incorporate not only the exact keywords but also the ones differing slightly due to minor typos, format inconsistencies, etc.;
2. designing an efficient and secure searching approach for file retrieval based on the resulted fuzzy keyword sets.

In this section, we will focus on the first part, i.e., building storage-efficient fuzzy keyword sets to facilitate the searching process.

4.1 Construction of Fuzzy Keyword Sets

To provide more practical and effective fuzzy keyword search constructions with regard to both storage and search efficiency, we now propose an advanced techniques to improve the straightforward approach for constructing the fuzzy keyword set. Without loss of generality, we will focus on the case of edit distance $d = 1$ to elaborate the proposed advanced techniques. For larger values of d, the reasoning is similar.

Gram-Based Fuzzy Set Construction. The gram of a string is a substring that can be used as a signature for efficient approximate search [26]. While gram has been widely used for constructing inverted list for approximate string search [27, 28, 29], we use gram for the matching purpose. We propose to utilize the fact that any primitive edit operation will affect at most one specific character of the keyword, leaving all the remaining characters untouched. In other words, the relative order of the remaining characters after the primitive operations is always kept the same as it is before the operations. With this significant observation, the fuzzy keyword set for a keyword w_i with ℓ single characters supporting edit distance d can be constructed as $S_{w_i,d} = \{S'_{w_i,\tau}\}_{0\leq\tau\leq d}$, where $S'_{w_i,\tau}$ consists of all the $(\ell–\tau)$-gram from w_i and with the same relative order (we assume that $d \leq \ell$). For example, the gram-based fuzzy set $S_{\texttt{CASTLE},1}$ for keyword `CASTLE` can be constructed as $\{\texttt{CASTLE}, \texttt{CSTLE}, \texttt{CATLE}, \texttt{CASLE}, \texttt{CASTE}, \texttt{CASTL}, \texttt{ASTLE}\}$. Generally, given a keyword w_i with ℓ single characters, the size of $S'_{w_i,\tau}$ is $C_\ell^{\ell-\tau}$, and the size of $S_{w_i,d}$ is $C_\ell^\ell + C_\ell^{\ell-1} + \cdots + C_\ell^{\ell-d}$. Compared to wildcard-based construction, gram-based construction can further reduce the storage of the index from 40MB to approximately 10MB under the same setting as in the wildcard-based approach.

4.2 The Intuitive Solutions

The size of fuzzy keyword set is greatly reduced using the proposed techniques. However, the above constructions introduce another challenge: How to generate the search request and how to perform fuzzy keyword search? In the straightforward approach, because the index is created by enumerating all of fuzzy words for each keyword, there always exist matching words for the search request as long as the edit distance between them is equal to or less than d. To design fuzzy search schemes based on the fuzzy keyword sets constructed from wildcard-based or gram-based technique, we compute the searching request regarding (w, k) as $\{T_{w'}\}_{w'\in S_{w,k}}$, where $S_{w,k} = \{S'_{w,0}, S'_{w,1}, \cdots, S'_{w,k}\}$ is generated in the same way as in the fuzzy keyword set construction. In this section, we will show how to achieve fuzzy keyword search based on the fuzzy sets constructed from the proposed advanced techniques. For simplicity, we will only consider the fixed d in our scheme designs. In this section, we start with some intuitive solutions, the analysis of which will motivate us to develop more efficient ones. Based on the storage-efficient fuzzy keyword set constructed as above, we first use the traditional listing approach to realize fuzzy keyword search supporting only one privilege and without user revocation.

Specifically, the scheme goes as follows:

1. In the index building phase, for each keyword $w_i \in W$, the data owner computes trapdoors $T_{w'_i} = f(sk, w'_i)$ for all $w'_i \in S_{w_i,d}$ with secret key sk. Then s/he computes $\mathsf{Enc}(sk, \mathrm{FID}_{w_i}\|w_i)$ and outsources the index table $\{\{T_{w'_i}\}_{w'_i\in S_{w_i,d}}, \mathsf{Enc}(sk, \mathrm{FID}_{w_i}\|w_i)\}$ together with the encrypted data files to the cloud server;
2. Assume an authorized user types in w as the searching input, with the pre-set

edit distance k. The searching input is first transformed to a fuzzy set $S_{w,k}$. Then, the trapdoors $\{T_{w'}\}_{w' \in S_{w,k}}$ for each element $w' \in S_{w,k}$ are generated and submitted as the search request to the cloud server;

3. Upon receiving the search request, the server first compares $\{T_{w'}\}_{w' \in S'_{w,0}}$ with the index and returns the search result as $\mathsf{Enc}(sk, \mathrm{FID}_w \| w)$ if there exists an exact match. Otherwise, the server will compare all the elements of $\{T_{w'}\}_{w' \in S'_{w,\tau}}$ $(1 \leq \tau \leq k)$ with the index for the file collection and return all of the matched results $\{\mathsf{Enc}(sk, \mathrm{FID}_{w_i} \| w_i)\}$.

The user now can obtain $\{\mathrm{FID}_{w_i} \| w_i\}$ through decryption and retrieve files of interest.

Another solution is to explore Bloom filter [30] to represent the fuzzy keyword set $S_{w_i,d}$ for each keyword w_i with edit distance d, namely, the trapdoor set $\{T_{w'_i}\}_{w'_i \in S_{w_i,d}}$ is inserted into keyword w_i's Bloom filter as the index stored on the server. Now by binding the encrypted file identifiers $\mathsf{Enc}(sk, \mathrm{FID}_{w_i} \| w_i)$ to w_i's Bloom filter, a per keyword index is generated to track the data files. Upon receiving the search request $\{T_{w'}\}_{w' \in S_{w,k}}$, the server tests which Bloom filters contain 1's in all r locations for each element $w' \in S_{w,k}$ and returns the search results, assuming there are r independent hash functions $h_1, \ldots, h_r$ used in the construction of Bloom filter. In this solution, the server will only need to store a bit array of m bits instead of the trapdoor information for all fuzzy set regarding w_i. Compared to the listing scheme, both storage cost and searching cost are now $O(|W|)$. The intuition behind this idea is to insert fuzzy set $S_{w_i,d}$ of all the keywords belonging to the same file into a single Bloom filter; a search request $\{T_{w'}\}_{w' \in S_{w,k}}$ for (w, k) is conducted by testing all the words in $\{T_{w'}\}_{w' \in S_{w,k}}$ through each file's Bloom filter. Note that the search cost associated with this solution is $O(|N|)$, where N is the number of data files.

4.3 Efficient Fuzzy Searching Scheme

In this section, we show how to improve the searching efficiency in multi-user setting, where a data owner stores a file collection on the cloud server and allows an arbitrary group of users with different privileges to search over his file collection. A broadcast encryption scheme is demanded in this paper. Assume $\mathsf{BE} = (\mathit{KeyGen}_{BE}, \mathit{Enc}_{BE}, \mathit{Dec}_{BE})$ is a broadcast encryption scheme providing revocation-scheme security against a coalition of revoked users [31]. More specially, KeyGen_{BE} is the key generation algorithm that is used to generate a long-lived key for the user, Enc_{BE} is the encryption algorithm that is used to encrypt files to a privileged user group G. The group G can be dynamically changing, as users can be added to or removed from G, Dec_{BE} is the decryption algorithm that is used to decrypt the ciphertext if with a non-revoked secret key at the time the message was encrypted.

We also need a secure symmetric encryption scheme in the following constructions. Assume SE=(KeyGen_{SE}, Enc_{SE}, Dec_{SE}) is a symmetric encryption scheme, where $KeyGen_{SE}$ is the setup algorithm with a predefined security parameter λ, Enc_{SE} and Dec_{SE} are the encryption and decryption algorithms, respectively.

Index Building. To enhance the searching efficiency, we utilize a symbol-based trie-traverse search scheme, where a multi-way tree is constructed for storing the fuzzy keyword set $\{S_{w_i,d}\}_{w_i \in W}$ over a finite symbol set. The key idea behind this construction is that all trapdoors sharing a common prefix may have common nodes. The root is associated with an empty set and the symbols in a trapdoor can be recovered in a search from the root to the leaf that ends the trapdoor. All fuzzy words in the trie can be found by a depth-first search. Assume $\Delta = \{\alpha_i\}$ is a predefined symbol set, where the number of different symbols is $\mid \Delta \mid = 2^n$, that is, each symbol $\alpha_i \in \Delta$ can be denoted by n bits.

The improved fuzzy keyword search scheme works as follows:

1. **Setup.** Assume the data owner wants to outsource the file collection $\mathcal{C}$ with keyword set W; s/he computes $T_{w'_i} = f(sk_P, w'_i)$ as $\alpha_{i_1} \cdots \alpha_{i_{l/n}}$ for each $w'_i \in S_{w_i,d}$ $(i = 1, \cdots, p)$ and each $P \in PS$, where l denotes the output length of one-way function $f(x)$, sk_P denotes the secret key related to the privilege P and PS denotes the privilege set defined by the data owner. A tree G_W covering all the fuzzy keywords of $w_i \in W$ is built up based on symbols in Δ. For each keyword W, the data owner determines its corresponding files $\{FID_i\}$ to be allowed to search under each eligible privilege P. The data owner encrypts F with the symmetric encryption as $C = Enc_{SE}(sk_P, F)$. Finally, the key sk_P is encapsulated with the encryption algorithm of CP-ABE under access policy P as $Enc_{ABE}(P, sk_P)$. The data owner chooses a random r and computes the broadcast encryption of r as $C' = Enc_{BE}(G, r)$ for the authorized user group G in this system. The data owner attaches the $\mathsf{Enc}(sk_P, \mathrm{FID}_{w_i} \| w_i)$ to G_W. The public parameter is then published on cloud servers so that every user can download it.
2. **New User Grant.** Assuming a new user wants to join the system, the data owner computes and sends the user a long-lived secret key of the broadcast encryption scheme. This key is to be used in user revocation. Assume that the user has been issued a private key sk_L on his/her attributes L by running the algorithm of $KeyGen(L, \cdot)$. Initially, the data owner wishes to allow users only with specific privileges to access the data files stored in cloud servers.
3. **Search.** To search files containing w with edit distance k, the eligible user first downloads the ciphertext C' of the broadcast encryption and decrypts to get r because the key r currently used is encrypted in the way that only the server and the set of currently authorized users can decrypt C' and get r. The user with the attribute private key on L can decrypt and get sk_P if his/her attribute list L matches the access policy P. Then, s/he computes $H(r, T')$ as a message authentication code (MAC), where T' is the requested trapdoor as $T' = \{T_{sk_P,w'}\}$ for each $w' \in S_{w,k}$. The user sends $T' = \{T_{sk_P,w'}\}_{w' \in S_{w,k}}$ and $H(r, T')$ to the server. Upon receiving T' and its MAC, the cloud servers verify its correctness

by using r. If it is valid, the cloud servers perform search. Specifically, the server divides each $T_{sk_P,w'}$ into a sequence of symbols, performs the search over G_W and returns $\{\mathsf{Enc}(sk_P, \mathrm{FID}_{w_i} \| w_i)\}$ to the user.

4. **User Revocation.** Whenever there is a user ID' to be revoked, the data owner picks a new r' and stores it encrypted on the cloud servers by computing $Enc_{BE}(G, r')$ such that only user set G, including the non-revoked users and cloud servers can decrypt it. Additionally, the original encrypted value $Enc_{BE}(G \cup ID', r)$ with respect to $G \cup ID'$ will be deleted. This broadcast encryption is utilized here to prevent the user who was authorized but currently revoked in the system.

   Note that by dividing the keying hash value into l/n parts, each n-bit of the hash value represents a symbol in Δ. The hash value of each fuzzy word $w_i' \in S_{w_i,d}$ is deterministic because with the same input sk and w_i', the output $\alpha_{i_1} \cdots \alpha_{i_{l/n}}$ is unique. Moreover, no information about w_i will be leaked from the output $\alpha_{i_1} \cdots \alpha_{i_{l/n}}$. In this scheme, the paths of trapdoors for different keywords are integrated by merging all the paths with the same prefix into a single trie to support more efficient search. The encrypted file identifiers will be indexed according to its address or name and the index information will be stored at the ending node of the corresponding path. With the returned search results $\{\mathsf{Enc}(sk_P, \mathrm{FID}_{w_i} \| w_i)\}$, the user may retrieve the files of his/her interest after decrypt and obtain $\{\mathrm{FID}_{w_i} \| w_i\}$. For each request, the search costs only $O(l/n)$ at the server side, which has nothing to do with the number of files or the size of related keywords.

5 SECURITY ANALYSIS

In this section, we analyze the correctness and security of the proposed fuzzy keyword search schemes. At first, we show the correctness of the schemes in terms of two aspects, that is, completeness and soundness.

Theorem 1. The scheme satisfies both completeness and soundness. Specially, upon receiving the request of w, all of the keywords $\{w_i\}$ will be returned if and only if $\mathsf{ed}(w, w_i) \leq k$.

The proof can be derived based on the following lemma:

Lemma 1. The intersection of the fuzzy sets $S_{w_i,d}$ and $S_{w,k}$ for w_i and w is not empty if and only if $\mathsf{ed}(w, w_i) \leq k$.

Proof. First, we show that $S_{w_i,d} \cap S_{w,k}$ is not empty when $\mathsf{ed}(w, w_i) \leq k$. To prove this, it is enough to find an element in $S_{w_i,d} \cap S_{w,k}$. Let $w = a_1 a_2 \cdots a_s$ and $w_i = b_1 b_2 \cdots b_t$, where all these a_i and b_j are single characters. After $\mathsf{ed}(w, w_i)$ edit operations, w can be changed to w_i according to the definition of edit distance. Let $w^* = a_1^* a_2^* \cdots a_m^*$, where $a_i^* = a_j$ or $a_i^* = *$ if any operation is performed at this position. Since the edit operation is inverted, from w_i, the same positions containing wildcard at w^* will be performed. Because $\mathsf{ed}(w, w_i) \leq k$, w^* is included in both $S_{w_i,d}$ and $S_{w,k}$, we get the result that $S_{w_i,d} \cap S_{w,k}$ is not empty.

Next, we prove that $S_{w_i,d} \cap S_{w,k}$ is empty if $\mathsf{ed}(w, w_i) > k$. The proof is given by reduction. Assume there exists an w^* belonging to $S_{w_i,d} \cap S_{w,k}$. We will show that $\mathsf{ed}(w, w_i) \leq k$, which reaches a contradiction. First, from the assumption that $w^* \in S_{w_i,d} \cap S_{w,k}$, we can get the number of wildcards in w^*, which is denoted by n^*, is not greater than k. Next, we prove that $\mathsf{ed}(w, w_i) \leq n^*$. We will prove the inequality with induction method. First, we prove it holds when $n^* = 1$. There are nine cases which should be considered: If w^* is derived from the operation of deletion from both w_i and w, then, $\mathsf{ed}(w_i, w) \leq 1$ because the other characters are the same except the character at the same position. If the operation is deletion from w_i and substitution from w, we have $\mathsf{ed}(w_i, w) \leq 1$ because they will be the same after at most one substitution from w_i. The other cases can be analyzed in a similar way and are omitted. Now, assuming that it holds when $n^* = \gamma$, we need to prove it also holds when $n^* = \gamma + 1$. If $\hat{w}^* = a_1^* a_2^* \cdots a_n^* \in S_{w_i,d} \cap S_{w,k}$, where $a_i^* = a_j$ or $a_i^* = *$. For a wildcard at position t, cancel the underlying operations and revert it to the original characters in w_i and w at this position. Assume two new elements w_i^* and w^* are derived from them, respectively. Then perform one operation at position t of w_i^* to make the character of w_i at this position be the same with w, which is denoted by w_i'. After this operation, w_i^* will be changed to w^*, which has only k wildcards. Therefore, we have $\mathsf{ed}(w_i', w) \leq \gamma$ from the assumption. We know that $\mathsf{ed}(w_i', w) \leq \gamma$ and $\mathsf{ed}(w_i', w_i) = 1$, based on which we know that $\mathsf{ed}(w_i, w) \leq \gamma + 1$. Thus, we can get $\mathsf{ed}(w, w_i) \leq n^*$. It renders the contradiction $\mathsf{ed}(w, w_i) \leq k$ because $n^* \leq k$. Therefore, $S_{w_i,d} \cap S_{w,k}$ is empty if $\mathsf{ed}(w, w_i) > k$. □

The following theorem says that in the gram-based search schemes, the satisfied keywords will be returned, as well as some keywords which are not desired. To be concrete, the returned keywords may also include keyword w_i as the answer for the request of (w, k) even if $\mathsf{ed}(w, w_i) > k$. For example, to search with the request $(\texttt{CAT}, 1)$, the keyword `CASTLE` will be returned if $(\texttt{CASTLE}, 3)$ is stored in the index, even if the edit distance of `CAT` and `CASTLE` is greater than 1. Thus, with the returned results, the user should filter the keyword set by further computing the edit distance.

Theorem 2. The gram-based fuzzy keyword search schemes satisfy the completeness. Specially, upon receiving the request of (w, k), all of the keywords w_i will be returned if $\mathsf{ed}(w, w_i) \leq k$.

The proof of the theorem can be obtained through the following lemma:

Lemma 2. Assume $S_{w_i,d}$ and $S_{w,k}$ are built with edit distance d and k for w_i and w, respectively. The set $S_{w_i,d} \cap S_{w,k}$ is not empty when $\mathsf{ed}(w_i, w) \leq \min\{d, k\}$.

Proof. Let $\mathsf{ed}(w_1, w_2) = d$, after at most d operations on the characters of w_1, it can be transformed to w_2. Without loss of generality, assume $|w_1| \geq |w_2|$. It means that the remaining $|w_1| - d$ characters in w_1 are untouched and they are equal to a $(|w_1| - d)$-character sequence in w_2, which belongs to $S_{w_1,k_1} \cap S_{w_2,k_2}$ when $d \leq \min\{k_1, k_2\}$. □

Theorem 3. The fuzzy keyword search schemes are secure regarding the search privacy.

Proof. In the gram-based scheme, the computation of index and request of the same keyword is identical. Therefore, we only need to prove the index privacy by using reduction. Suppose the searchable encryption scheme fails to achieve the index privacy against the indistinguishability under the chosen keyword attack, which means there exists an algorithm $\mathcal{A}$ which can get the underlying information of keyword from the index. Then, we build an algorithm $\mathcal{A}'$ that utilizes $\mathcal{A}$ to determine whether some function $f'(\cdot)$ is a pseudo-random function such that $f'(\cdot)$ is equal to $f(sk, \cdot)$ or a random function. $\mathcal{A}'$ has access to an oracle $\mathcal{O}_{f'(\cdot)}$ that takes as input secret value x and returns $f'(x)$. Upon receiving any request of the index computation, $\mathcal{A}'$ answers it with request to the oracle $\mathcal{O}_{f'(\cdot)}$. After making these trapdoor queries, the adversary outputs two challenge keywords w_0^* and w_1^* with the same length and edit distance, which can be relaxed by adding some redundant trapdoors. $\mathcal{A}'$ picks one random $b \in \{0,1\}$ and sends w_b^* to the challenger. Then, $\mathcal{A}'$ is given a challenge value y, which is either computed from a pseudo-random function $f(sk, \cdot)$ or a random function. $\mathcal{A}'$ sends y back to $\mathcal{A}$, which answers with $b' \in \{0,1\}$. Suppose $\mathcal{A}$ guesses b correctly with non-negligible probability, which indicates that the value is not randomly computed. Then, $\mathcal{A}'$ makes a decision that $f'(\cdot)$ is a pseudo-random function. As a result, based on the assumption of the indistinguishability of the pseudo-random function from some real random function, $\mathcal{A}$ at best guesses b correctly with approximate probability 1/2. Thus, the search privacy is obtained. □

Theorem 4. The fuzzy keyword search schemes are secure regarding the search authorization.

Proof. To prove the security against the users with unauthorized search, we need to prove that, given access to a set of encrypted data, the attackers are not able to learn any partial information about the underlying trapdoors of any keyword. Recall that there are three kinds of attackers here: 1) cloud servers; 2) revoked users; 3) users with unmatched attributes.

- For revoked users, it is easy to prove the security because they cannot get new r' encrypted by the broadcast encryption with respect to eligible users; thus, they cannot generate valid trapdoor.
- For cloud servers, they have not attribute private key on any attribute set; that is, they cannot get any information of the secret key used in the symmetric encryption. Therefore, they are not able to decrypt and get any information of the underlying files if the symmetric encryption is secure.

- For users whose attributes do not match the access policy, they cannot get the underlying secret key used in the symmetric encryption. As a result, they cannot decrypt to get the files though they have r'.
- If they collude, based on the security of the original ABE, it already has been designed to prevent from such attack.

Based on the above analysis, the key point of the security proof is the underlying fine-grained ABE. We only need to prove that the ABE will not leak information of the underlying message. The formal definition and secure scheme has been given in [24], which is called indistinguishability against chosen message attack. More specially, after setup of the system and uploading of the data, the adversary submits some queries to retrieve the attribute private keys. Finally, $\mathcal{A}$ cannot distinguish if a ciphertext is an encryption to which one of two adaptively chosen messages under some chosen access policy. □

6 CONCLUSION

In this paper, the fuzzy keyword search in a multi-user system with differential privileges is addressed. We formalized and solved the problem of supporting efficient yet privacy-preserving fuzzy search for achieving effective utilization of remotely stored encrypted data in cloud computing. We utilized the gram-based technique to construct the storage-efficient fuzzy keyword sets by exploiting the similarity metric of edit distance. Based on the constructed fuzzy keyword sets, we further proposed a brand new symbol-based trie-traverse searching scheme, where a multi-way tree structure is built up using symbols transformed from the resulted fuzzy keyword sets. Through rigorous security analysis, we showed that our proposed solution is secure and privacy-preserving, while correctly realizing the goal of fuzzy keyword search.

Acknowledgements

This work is partially supported by the National Natural Science Foundation of China (Nos. 61100224 and 61272455), Grant for Universities in Guangzhou City (Grant No. 10A008), the Foundation for Distinguished Young Talents in Higher Education of Guangdong Province (Grant No. LYM10106), and China 111 Project (No. B08038).

REFERENCES

[1] Korkmaz, T.—Tek, S.: Analyzing Response Time of Batch Signing. Journal of Internet Services and Information Security, Vol. 1, 2011, No. 1, pp. 70–85.

[2] Fukushima, K.—Kiyomoto, S.—Miyake, Y.: Towards Secure Cloud Computing Architecture – A Solution Based on Software Protection Mechanism. Journal of Internet Services and Information Security, Vol. 1, 2011, pp. 4–17.

[3] Lu, Y.—Tsudik, G.: Privacy-Preserving Cloud Database Querying. Journal of Internet Services and Information Security, Vol. 1, No. 4, pp. 5–25.

[4] Shiraishi, Y.—Mohri, M.—Fukuta, Y.: A Server-Aided Computation Protocol Revisited for Confidentiality of Cloud Service. Journal of Wireless Mobile Networks, Ubiquitous Computing and Dependable Applications, Vol. 2, 2011, No. 2, pp. 83–94.

[5] Pahl, C.—Zhu, Y.: Data Integration in Mediated Service Compositions. Computing and Informatics, Vol. 31, 2012, No. 6, pp. 1129–1149.

[6] Lee, M.—Cho, N.—Lee, K.—Ko, K.: Design and Implementation of an Intranet Security and Access Control System in Ubi-Com. Computing and Informatics, Vol. 30, 2011, No. 3, pp. 419–428.

[7] Li, J.—Wang, Q.—Wang, C.—Cao, N.—Ren, K.: Fuzzy Keyword Search over Encrypted Data in Cloud Computing. Proceedings of the 29th IEEE INFOCOM 2010.

[8] Li, J.—Wang, Q.—Wang, C.—Cao, N.—Ren, K.—Lou, W.: Enabling Efficient Fuzzy Keyword Search over Encrypted Data in Cloud Computing. Available at `http://eprint.iacr.org/2009/593.pdf`.

[9] Li, J.—Li, J.—Chen, X.—Jia, C.—Liu, Z.: Efficient Keyword Search over Encrypted Data with Fine-Grained Access Control in Hybrid Cloud. Proc. of NSS 2012, pp. 490–502.

[10] Li, J.—Jia, C.—Li, J.—Liu, Z.: A Novel Framework for Outsourcing and Sharing Searchable Encrypted Data on Hybrid Cloud. Proc. of INCoS 2012, pp. 1–7.

[11] Bellare, M.—Boldyreva, A.—O'Neill, A.: Deterministic and Efficiently Searchable Encryption. Proceedings of Crypto 2007, LNSC, Vol. 4622, Springer-Verlag 2007.

[12] Song, D.—Wagner, D.—Perrig, A.: Practical Techniques for Searches on Encrypted Data. Proc. of IEEE Symposium on Security and Privacy 2000.

[13] Goh, E.-J.: Secure Indexes. Cryptology ePrint Archive, Report 2003/216, 2003, `http://eprint.iacr.org/`.

[14] Boneh, D.—Crescenzo, G.D.—Ostrovsky, R.—Persiano, G.: Public Key Encryption With Keyword Search. Proc. of EUROCRYP 2004.

[15] Waters, B.—Balfanz, D.—Durfee, G.—Smetters, D.: Building an Encrypted and Searchable Audit Log. Proc. of 11th Annual Network and Distributed System 2004.

[16] Chang, Y.-C.—Mitzenmacher, M.: Privacy Preserving Keyword Searches on Remote Encrypted Data. Proc. of ACNS 2005.

[17] Curtmola, R.—Garay, J. A.—Kamara, S.—Ostrovsky, R.: Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions. Proc. of ACM CCS 2006.

[18] Boneh, D.—Waters, B.: Conjunctive, Subset, and Range Queries on Encrypted Data. Proc. of TCC 2007, pp. 535–554.

[19] Bao, F.—Deng, R.—Ding, X.—Yang, Y.: Private query on encrypted data in multi-user settings. Proc. of ISPEC 2008.

[20] Shi, E.—Bethencourt, J.—Chan, T.-H.—Song, D.—Perrig, A.: Multidimensional Range Query over Encrypted Data. IEEE Symposium on Security and Privacy 2007.

[21] Feigenbaum, J.—Ishai, Y.—Malkin, T.—Nissim, K.—Strauss, M.—Wright, R. N.: Secure Multiparty Computation of Approximations. Proc. of ICALP '01.

[22] Beimel, K. N. A.—Carmi, P.—Weinreb, E.: Private Approximation of Search Problems. Proc. of 38th Annual ACM Symposium on the Theory of Computing 2006, pp. 119–128.

[23] Ostrovsky, R.: Software Protection and Simulations on Oblivious RAMs. Ph. D. dissertation, Massachusetts Institute of Technology 1992.

[24] Goyal, V.—Pandey, O.—Sahai, A.—Waters, B.: Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data. CCS '06, ACM 2006, pp. 89–98.

[25] Levenshtein, V.: Binary Codes Capable of Correcting Spurious Insertions and Deletions of Ones. Problems of Information Transmission, Vol. 1, 1965, No. 1, pp. 8–17.

[26] Ji, S.—Li, G.—Li, C.—Feng, J.: Efficient Interactive Fuzzy Keyword Search. Proc. of WWW 2009.

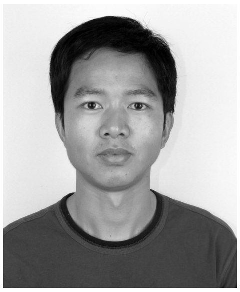
[27] Chaudhuri, S.—Ganti, V.—Motwani, R.: Robust Identification of Fuzzy Duplicates. Proc. of ICDE 2005.

[28] Arasu, A.—Ganti, V.—Kaushik, R.: Efficient Exact set-Similarity Joins. Proc. of VLDB 2006, pp. 918–929.

[29] Chakrabarti, K.—Chaudhuri, S.—Ganti, V.—Xin, D.: An Efficient Filter for Approximate Membership Checking. Proc. of SIGMOD 2006.

[30] Bloom, B.: Space/Time Trade-Offs in Hash Coding with Allowable Errors. Communications of the ACM, Vol. 13, 1970, No. 7, pp. 422–426.

[31] Fiat, A.—Naor, M.: Broadcast Encryption. Proc. of CRYPTO 1993.



Jin Li works at Guangzhou University. He received his B. Sc. (2002) and M. Sc. (2004) from Southwest University and Sun Yat-sen University, both in mathematics. He got his Ph. D. degree in information security from Sun Yat-sen University in 2007. His research interests include applied cryptography and security in cloud computing (secure outsourcing computation and cloud storage).

Xiaofeng Chen works at Xidian University as a Professor. He received his B. Sc. and M. Sc. in mathematics from Northwest University, China, and his Ph. D. degree in cryptography from Xidian University in 2003. His research interests include applied cryptography and cloud security.