Keyword based Data Indexing and Receiving in Protected Cloud Environment

Shrthi S V
Mtech IV Semester
Kalpataru Institute of Technology, Tiptur

Jagadeesha R
Assistant Professor Dept. of CSE
Kalpataru Institute of Technology, Tiptur

***Abstract*:- A Keyword Based Data Indexing and Receiving In Protected Cloud Environment due to the increasing popularity of cloud computing, more and more data owners are motivated to outsource their data to cloud servers for great convenience and reduced cost in data management. However, sensitive data should be encrypted before outsourcing for privacy requirements, which obsoletes data utilization like keyword-based document retrieval. In this paper, we present a secure multi-keyword ranked search scheme over encrypted cloud data, which simultaneously supports dynamic update operations like deletion and insertion of documents. Specifically, the vector space model and the widely-used tf idf model are combined in the index construction and query generation. We construct a special tree-based index structure and propose a "greedy depth-first search" algorithm to provide efficient multi-keyword ranked search. The secure knn algorithm is utilized to encrypt the index and query vectors, and meanwhile ensure accurate relevance score calculation between encrypted index and query vectors. In order to resist statistical attacks, phantom terms are added to the index vector for blinding search results. Due to the use of our special tree-based index structure, the proposed scheme can achieve sub-linear search time and deal with the deletion and insertion of documents flexibly. Extensive experiments are conducted to demonstrate the efficiency of the proposed scheme.**

I. INTRODUCTION

Cloud computing has been considered as a new model of enterprise it infrastructure, which can organize huge resource of computing, storage and applications, and enable users to enjoy ubiquitous, convenient and on-demand network access to a shared pool of configurable computing resources with great efficiency and minimal economic overhead [1]. Attracted by these appealing features, both individuals and enterprises are motivated to outsource their data to the cloud, instead of purchasing software and hardware to manage the data themselves.

Despite of the various advantages of cloud services, outsourcing sensitive information (such as e-mails, personal health records, company finance data, government documents, etc.)

To remote servers brings privacy concerns. The cloud service providers (csps) that keep the data for users may access users' sensitive information without authorization. A general approach to protect the data confidentiality is to encrypt the data before outsourcing [2]. However, this will cause a huge cost in terms of data usability. For example, the existing techniques on keyword-based information retrieval, which are widely used on the plaintext data, cannot be directly applied on the encrypted data. Downloading all the data from the cloud and decrypt locally is obviously impractical.

In order to address the above problem, researcher-s have designed some general-purpose solutions with fully homomorphism encryption [3] or oblivious rams [4]. However, these methods are not practical due to their high computational overhead for both the cloud sever and user. On the contrary, more practical special-purpose solutions, such as searchable encryption (se) schemes have made specific contributions in terms of efficiency, functionality and security. Searchable encryption schemes enable the client to store the encrypted data to the cloud and execute keyword search over cipher text domain. So far, abundant works have been proposed under different threat models to achieve various search functionality, such as single keyword search, similarity search, multi-keyword boolean search, ranked search, multi-keyword ranked search, etc. Among them, multi-keyword ranked search achieves more and more attetion for its practical applicability. Recently, some *dynamic* schemes have been proposed to support inserting and deleting operations on document collection. These are significant works as it is highly possible that the data owners need to update their data on the cloud server. But few of the dynamic schemes support efficient multi-keyword ranked search.

This paper proposes a secure tree-based search scheme over the encrypted cloud data, which supports multi-keyword ranked search and dynamic operation on the document collection. Specifically, the vector space model and the widely-used "term frequency (tf) × inverse document frequency (idf)" model are combined in the index construction and query generation to provide multi-keyword ranked search.

In order to obtain high search efficiency, we construct a tree-based index structure and propose a "greedy depth-first search" algorithm based on this index tree. Due to the special structure of our tree-based index, the proposed search scheme can flexibly achieve sub-linear search time and deal with the deletion and insertion of documents. The secure knn algorithm is utilized to encrypt the index and query vectors, and meanwhile ensure accurate relevance score calculation between encrypted index and query vectors. To resist different attacks in different threat models, we construct two secure search schemes: the basic dynamic multi-keyword ranked search (bdmrs) scheme in the known cipher text model, and the enhanced dynamic multi-keyword ranked search (edmrs) scheme in the known background model. Our contributions are summarized as follows:

1) we design a searchable encryption scheme that supports both the accurate multi-keyword ranked search and flexible dynamic operation on document collection.
2) due to the special structure of our tree-based index, the search complexity of the proposed scheme is fundamentally kept to logarithmic. And in practice, the proposed scheme can achieve higher search efficiency by executing our "greedy depth-first search" algorithm. Moreover, parallel search can be flexibly performed to further reduce the time cost of search process.

The reminder of this paper is organized as follows. Related work is discussed in section 2, and section 3 gives a brief introduction to the system model, threat model, the design goals, and the preliminaries. Section 4 describes the schemes in detail. Section 5 presents the experiments and performance analysis. And section 6 covers the conclusion.

II. RELATED WORK

Searchable encryption schemes enable the clients to store the encrypted data to the cloud and execute keyword search over cipher text domain. Due to different cryptography primitives, searchable encryption schemes can be constructed using public key based cryptography [5], [6] or symmetric key based cryptography [7], [8], [9], [10].

Song *et al.* [7] proposed the first symmetric searchable encryption (sse) scheme, and the search time of their scheme is linear to the size of the data collection. Goh [8] proposed formal security definitions for sse and designed a scheme based on bloom filter. The search time of goh's scheme is $o(n)$, where n is the cardinality of the document collection. Curtmola *et al.* [10] proposed two schemes (sse-1 and sse-2) which achieve the optimal search time. Their sse-1 scheme is secure against chosen-keyword attacks (cka1) and sse-2 is secure against adaptive chosen-keyword attacks (cka2). These early works are single keyword boolean search schemes, which are very simple in terms of functionality. Afterward, abundant works have been proposed under different threat models to achieve various search functionality, such as single keyword search, similarity search [11], [12], [13], [14], multi-keyword boolean search [15], [16], [17], [18], [19], [20], [21], [22], ranked search [23], [24], [25], and multi-keyword ranked search [26], [27], [28], [29], etc.

Multi-keyword boolean search allows the users to input multiple query keywords to request suitable documents. Among these works, conjunctive keyword search schemes [15], [16], [17] only return the documents that contain all of the query keywords. Disjunctive keyword search schemes [18], [19] return all of the documents that contain a subset of the query keywords. Predicate search schemes [20], [21], [22] are proposed to support both conjunctive and disjunctive search. All these multi-keyword search schemes retrieve search results based on the existence of keywords, which cannot provide acceptable result ranking functionality.

Ranked search can enable quick search of the most relevant data. Sending back only the top-k most relevant documents can effectively decrease network traffic. Some early works [23], [24], [25] have realized the ranked search using order-preserving techniques, but they are designed only for single keyword search. Cao *et al.* [26] realized the first privacy-preserving multi-keyword ranked search scheme, in which documents and queries are represented as vectors of dictionary size. With the "coordinate matching", the documents are ranked ac-cording to the number of matched query keywords. However, cao *et al.*'s scheme does not consider the importance of the different keywords, and thus is not accurate enough. In addition, the search efficiency of the scheme is linear with the cardinality of document collection. Sun *et al.* [27] presented a secure multi-keyword search scheme that supports similarity-based ranking.

The authors constructed a searchable index tree based on vector space model and adopted cosine measure together with tf×idf to provide ranking results. Sun *et al.*'s search algorithm achieves better-than-linear search efficiency but results in precision loss. Orencik *et al.* [28] proposed a secure multi-keyword search method which utilized local sensitive hash (lsh) functions to cluster the similar documents. The lsh algorithm is suitable for similar search but cannot provide exact ranking. In [29], zhang *et al.* Proposed a scheme to deal with secure multi-keyword ranked search in a multi-owner model. In this scheme, different data owners use different secret keys to encrypt their documents and keywords while authorized data users can query without knowing keys of these different data owners. The authors proposed an "additive order preserving function" to retrieve the most relevant search results. However, these works don't support dynamic operations.

Practically, the data owner may need to update the document collection after he upload the collection to the cloud server. Thus, the se schemes are expected to sup-port the insertion and deletion of the documents. There are also several dynamic searchable encryption schemes. In the work of song *et al.* [7], the each document is considered as a sequence of fixed length words, and is individually indexed. This scheme supports straight-forward update operations but with low efficiency. Goh [8] proposed a scheme to generate a sub-index (bloom filter) for every document based on keywords. Then the dynamic operations can be easily realized through updating of a bloom filter along with the corresponding document.

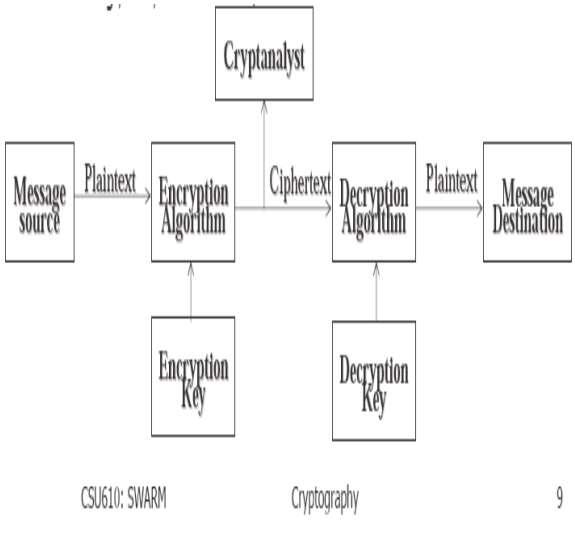
However, goh's scheme has linear search time and suffers from false positives. In 2012, kamara *et al.* [30] constructed an encrypted inverted index that can handle dynamic data efficiently. But, this scheme is very complex to implement. Subsequently, as an im-provement, kamara *et al.* [31] proposed a new search scheme based on tree-based index, which can handle dynamic update on document data stored in leaf n-odes. However, their scheme is designed only for single-keyword boolean search. In [32], cash *et al.* Presented a data structure for keyword/identity tuple named "t-set". Then, a document can be represented by a series of independent t-sets. Based on this structure, cash *et al.* [33] proposed a dynamic searchable encryption scheme. In their construction, newly added tuples are stored in another database in the cloud,

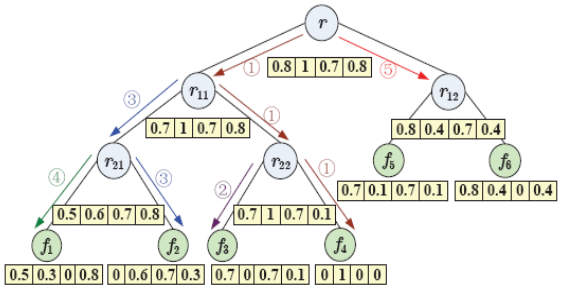
and deleted tuples are recorded in a revocation list. The final search result is achieved through excluding tuples in the revocation list from the ones retrieved from original and newly added tuples. Yet, cash *et al.*'s dynamic search scheme doesn't realize the multi-keyword ranked search functionality.

III EXISTING SYSTEM

In the existing techniques on keyword-based information retrieval, which are widely used on the plaintext data, cannot be directly applied on the encrypted data. Downloading all the data from the cloud and decrypt locally is obviously impractical. All these multi keyword search schemes retrieve search results based on the existence of keywords, which cannot provide acceptable result ranking functionality. However, sensitive data should be encrypted before outsourcing for privacy requirements, which obsoletes data utilization like keyword-based document retrieval.

Existing system algorithms





Specifically, the vector space model and the widely-used tf×df model are combined in the index construction and query generation.

A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data we construct a special tree-based index structure and propose a "greedy depth-first search" algorithm to provide efficient multi-keyword ranked search. The proposed scheme can achieve sub-linear search time and deal with the deletion and insertion of documents flexibly. Extensive experiments are conducted to demonstrate the efficiency of the proposed scheme.

- Abundant works have been proposed under different threat models to achieve various search functionality,
- Recently, some dynamic schemes have been proposed to support inserting and deleting operations on document collection.
- This paper proposes a secure tree-based search scheme over the encrypted cloud data, which supports multi keyword ranked search and dynamic operation on the document collection.

Proposed system algorithms

- Algorithm to provide efficient multi-keyword ranked search .
- The secure knn algorithm is utilized to encrypt the index and query vectors.
- Propose a "greedy depth-first search" algorithm based on this index tree.
- Algorithm achieves better-than-linear search efficiency but results in precision loss.
- The lsh algorithm is suitable for similar search but cannot provide exact ranking.
- {i′s ; ci} ← genupdateinfo (sk; ts; i; up type)) this algorithm generates the update information {i′s ; ci} which will be sent to the cloud server.

Advantages

Despite of the various advantages of cloud services, outsourcing sensitive information such as e-mails, personal health records, company finance data, government documents, etc.

System architecture

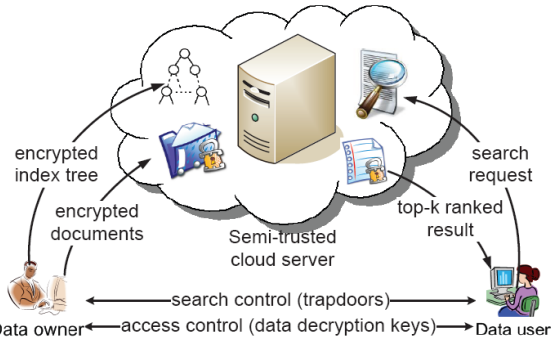


Fig. 1. The architecture of ranked search over encrypted cloud data

Tree-based index with the document collection

We construct a special keyword balanced binary tree as the index, and propose a "greedy depth-first search" algorithm to obtain better efficiency than linear search.

- Data owner
- Data user
- Dynamic multi-keyword ranked search
- Search efficiency

- Privacy-preserving:
  - *Index confidentiality and query confidentiality*
  - *Trapdoor unlinkability*:
  - *Keyword privacy*
- Dynamic update operation

Algorithm:

- Edmrs scheme
- Tree-based search algorithm:
- Secure scheme

- ✓ Our proposed search scheme achieves multi-keyword ranked search over encrypted data with high efficiency and search result accuracy.
- ✓ We propose a secure dmrs scheme which meets privacy requirements in the known ciphertext model.
- ✓ Benefiting from tree-based index structure, our search scheme supports dynamic update operation (like deletion and insertion) on documents, which caters to real-world needs and is superior to most current static schemes.

To enable efficient, secure and dynamic multi-keyword ranked search over outsourced encrypted cloud data under the aforementioned models, our system design should simultaneously achieve the following design goals.

1) Dynamic multi-keyword ranked search: to design a search scheme over encrypted data which provides not only effective multi-keyword query and accurate result ranking, but also dynamic update on document collections.
2) Search efficiency: our search scheme aims to achieve better practical search efficiency than linear search [13] by exploring a tree-based index structure and an efficient search algorithm.
3) Privacy-preserving: to prevent the cloud server from learning additional information from the dataset, the index tree, and the queries. The specific search privacy requirements are summarized as follows,
1) index confidentiality and query confidentiality: the underlying plaintext information (including keywords in the index and query, keywords' tf values stored in the index, and idf values of query keywords) should be protected from cloud server;
2) trapdoor unlinkability: the cloud server should not be able to determine whether two encrypted queries (trapdoors) are generated from the same search request;
3) keyword privacy: the cloud server could not identify the specific keyword in query, index or dataset.
Search algorithm

The search process of our dmrs scheme starts from the root node with a recursive procedure upon the tree in a special depth-first manner, which is called as "greedy depth-first traverse strategy". Specifically, if the node's similarity score is less than or equal to the minimum similarity score of the currently selected top-documents, search process returns to the parent node, otherwise, it goes down to examine the child node. The similarity score of each node is calculated as formula (1), *i.e.*, the inner product of query vector and data vector. This procedure is executed recursively until the objects with top- scores are selected. The search can be done very efficiently, since only part of the index tree is visited due to the relatively accurate maximum score prediction.

1) Index confidentiality and query confidentiality: in dmrs, and are obfuscated vectors, which means the cloud server cannot infer the original vectors or without the secret key. Therefore, index confidentiality and query confidentiality are well protected.

2) Query unlinkability: the trapdoor of query vector is generated from random splitting operation, which means same search requests would be transformed into different query vectors (trapdoors), therefore, the query unlinkability is protected. However, equipped with capability on tracking visited nodes with corresponding similarity scores, the cloud server might be able to link the same search requests according to the same similarity scores. Under this circumstance, the query unlinkability is unavailable.
3) Keyword privacy: in the known ciphertext model, the cloud server is supposed to only know the encrypted document set, index tree and trapdoor. Therefore, without other background information, the cloud server is unable to deduce keywords or tf/idf values from the result similarity scores. However, in enhanced threat model, the cloud server may be equipped with more knowledge like document/keyword frequency statistics of the dataset. Then the cloud server could launch statistical attack to deduce or even identify specific keywords in the query. As an improvement, our future work aims to design a secure scheme that meets all the privacy requirements above even in enhanced threat model.

Dynamic update operation
Since our dmrs scheme is designed on a red-black tree data structure, the dynamic operations (like insertion or deletion of a document) could be executed efficiently through structural update on the index tree. Furthermore, since the documents are directly related to the leaf nodes, the whole structure of index tree would change little

V CONCLUSION AND FUTURE WORK

In this paper, a secure, efficient and dynamic search scheme is proposed, which supports not only the accurate multi-keyword ranked search but also the dynamic deletion and insertion of documents. We construct a special keyword balanced binary tree as the index, and propose a "greedy depth-first search" algorithm to obtain better efficiency than linear search. In addition, the parallel search process can be carried out to further reduce the time cost. The security of the scheme is protected against two threat models by using the secure knn algorithm. Experimental results demonstrate the efficiency of our proposed scheme.

There are still many challenge problems in symmetric se schemes. In the proposed scheme, the data owner is responsible for generating updating information and sending them to the cloud server. Thus, the data owner needs to store the unencrypted index tree and the information that are necessary to recalculate the idf values.

Such an active data owner may not be very suitable for the cloud computing model. It could be a meaningful but difficult future work to design a dynamic searchable encryption scheme whose updating operation can be completed by cloud server only, meanwhile reserving the ability to support multi-keyword ranked search. In addition, as the most of works about searchable encryption, our scheme mainly considers the challenge from the cloud server. Actually, there are many secure challenges in a multi-user scheme. Firstly, all the users usually keep the same secure key for trapdoor generation in a symmetric se scheme.

In this case, the revocation of the user is big challenge. If it is needed to revoke a user in this scheme, we need to rebuild the index and distribute the new secure keys to all the authorized users. Secondly, symmetric se schemes usually assume that all the data users are trustworthy. It is not practical and a dishonest data user will lead to many secure problems. For example, a dishonest

Data user may search the documents and distribute the decrypted documents to the unauthorized ones. Even more, a dishonest data user may distribute his/her secure keys to the unauthorized ones. In the future works, we will try to improve the se scheme to handle these challenging problems.

REFERENCES

[1] K. Ren, C.Wang, Q.Wang et al., "Security challenges for the public cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69–73, 2012.

[2] S. Kamara and K. Lauter, "Cryptographic cloud storage," in Financial Cryptography and Data Security. Springer, 2010, pp. 136–149.

[3] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. Dissertation, Stanford University, 2009.

[4] O. Goldreich and R. Ostrovsky, "Software protection and simulation on oblivious rams," Journal of the ACM (JACM), vol. 43, no. 3, Pp. 431–473, 1996.

[5] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in Advances in Cryptology- Eurocrypt 2004. Springer, 2004, pp. 506–522.

[6] D. Boneh, E. Kushilevitz, R. Ostrovsky, and W. E. Skeith III, "Public key encryption that allows pir queries," in Advances in Cryptology-CRYPTO 2007. Springer, 2007, pp. 50–67.

[7] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on. IEEE, 2000, pp. 44–55.

[8] E.-J. Goh et al., "Secure indexes." IACR Cryptology eprint Archive, vol. 2003, p. 216, 2003.

[9] Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in Proceedings of the Third international Conference on Applied Cryptography and Network Security. Springer-Verlag, 2005, pp. 442–455.

[10] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," In Proceedings of the 13th ACM conference on Computer and communications security. ACM, 2006, pp. 79–88.

[11] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in INFOCOM, 2010 Proceedings IEEE. IEEE, 2010, pp. 1–5.

[12] M. Kuzu, M. S. Islam, and M. Kantarcioglu, "Efficient similarity search over encrypted data," in Data Engineering (ICDE), 2012 IEEE 28th International Conference on. IEEE, 2012, pp. 1156–1167.

[13] C. Wang, K. Ren, S. Yu, and K. M. R. Urs, "Achieving usable and privacy-assured similarity search over outsourced cloud data," in INFOCOM, 2012 Proceedings IEEE. IEEE, 2012, pp. 451–459. 1045-9219 (c) 2015 IEEE. Personal use is permitted,