DYNAMIC SEARCHABLE ENCRYPTION FOR TOPK QUERIES IN CLOUD

[#1]S.RAJESH, M.Tech Student,

[#2]N.MAHESH, Associate Professor,

[#3]Dr.M.SUJATHA, Associate Professor,

Department Of CSE,

JYOTHISHMATHI INSTITUTE OF TECHNOLOGICAL SCIENCES, KARIMNAGAR T.S.INDIA.

ABSTRACT: Searchable symmetric encryption (SSE) enables a client to perform searches over its outsourced encrypted files while preserving privacy of the files and queries. Dynamic schemes, where files can be added or removed, leak more information than static schemes. For dynamic schemes, forward privacy requires that a newly added file cannot be linked to previous searches. We present a new dynamic SSE scheme that achieves forward privacy by replacing the keys revealed to the server on each search. 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.

***Keywords**: Searchable encryption, multi-keyword ranked search, dynamic update, cloud computing.*

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. 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. It is now common practice for an individual to share or backup documents using cloud storage services. Examples of services include Google Drive, Microsoft OneDrive, Apple iCloud, Dropbox and Amazon S3. This provides an individual with easy access to his/her data anywhere and anytime. In addition, more and more organizations from both the private and government sectors are moving their data and computations to the cloud. In the US, for instance, both Amazon and Microsoft provide specialized cloud services called AWS GovCloud and Azure Government to government agencies. These services are physically isolated from the regular cloud services offered by both providers and need to adhere to strict governmen t security regulations. The existence of such services is probably attributable, in part, to the surge in usage of cloud-based services in the government sector. According to a report by security audit firm Netwrix, the number of cloud adoptions in the surveyed organizations increased from 43% (2015) to 68% (2016) [1]. The report also states that cloud security is a primary concern for 70% of organizations worldwide. In a report published in 2016, Cisco anticipates that by the year 2020, 59% of the world's Internet users will use personal cloud storage, which is an increase from 47% in 2015 [2]. These trends are not uncommon since a cloud storage service brings numerous benefits to the user. The service, however, is not without its downsides.

One common property inherited by all the current cloud services is that the user does not have total control of the privacy of the stored documents. This is inevitable since the provider is expected to have read access privileges in order to be able to search through documents related to the user's query. A seemingly trivial solution would be for the user to encrypt all documents prior to storing them on the cloud. However, the user now loses the ability to search these documents. The workaround is either to download and decrypt all documents locally or give the encryption key to the cloud provider. The former approach is extremely inefficient while the latter raises privacy concerns. In order to protect the privacy of documents while at the same time allowing a user to efficiently search them, we introduce a system based on our multi-server SSE scheme that we call the searchable data vault (SDV). The SDV enables a user to store encrypted documents in the cloud and to retrieve them in an efficient and privacy-preserving manner. Our main contributions are as follows:

1. SDV allows documents, or pieces of an encrypted document, called blocks, to be stored in different cloud storage services, in contrast to existing schemes and systems that focus on outsourcing to a single storage. It ensures with high probability that no single storage provider has a complete set of blocks, which it may use to learn additional information about the document. As far as we know, it is the first searchable encryption system that provides such a feature.
2. A core component of SDV is a controller that manages query indexes, document submission, and retrieval. The controller is designed in such a way that the underlying searchable symmetric encryption (SSE) scheme, utilized for efficient search, is "pluggable". It means existing schemes that cater for single storage provider can be adapted for use with our system. The controller may be implemented and placed in a query gateway. This is the case for our deployment.
3. We further propose a multi-server SSE scheme, which is adapted from [3], with an implementation for SDV.
4. The query index is designed to cater to a two-level dictionary structure, where the basic level is for a single-word query. An upper level can be included for future extension to more expressive queries (e.g., ranked, range, conjunctive).

Figure 1 illustrates the architecture of SDV, which consists of three main modules that provide the mechanisms contributing to our main results:

• SDV app: This module provides an interface for the SDV system to interact with the users for document upload and retrieval. In our deployment, it is also integrated with a unified authentication platform [4] for the purpose of user registration and authentication.

• SDV controller: This module maintains a database that contains credentials of users and cloud servers. It also creates a query index table, and manages the submission and retrieval of encrypted documents.

• SDV API: This module contains an implementation of our multi-server SSE scheme. The scheme divides and encrypts blocks of documents, and generates index entries for query purposes.

Figure 2 further illustrates our modular approach in designing the SDV controller, so that the underlying searchable encryption scheme can be replaced without too much modification in the system, as well as providing flexible extension to various search functionalities. The idea is to obtain only the index entries from the searchable encryption scheme. Creation of the query index table is done at the SDV controller, instead of the common approach of creating the index table by the SSE scheme. As for search functionalities, our intuition is to maintain a basic query index table with filename as the keyword token. This token links to the encrypted blocks and the servers that store these blocks. By doing so, query module based on information retrieval techniques, or expressive searchable encryption mechanisms, for example in [5,6] can be customized to return filenames matching the query keywords. These filenames can then be passed to the basic index table for document retrieval. We discuss our multi-server SSE scheme utilizing the filename as a keyword in Section 4.

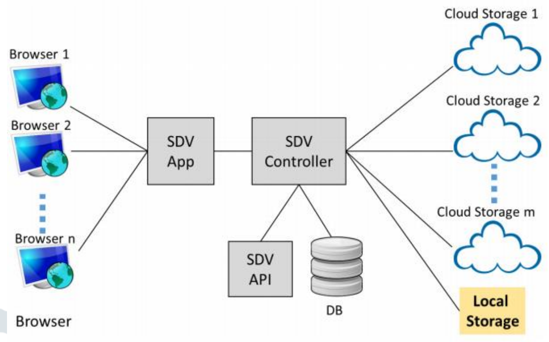

Figure 1. The searchable data vault (SDV)

The searchable data vault (SDV) consists of three main components: SDV application, SDV controller, and SDV API. Users access the system through a web browser. The SDV app provides the interface to interact with the users. The SDV controller manages the credentials database, and the core functionalities through interaction with the SDV API, which consists of the searchable symmetric encryption (SSE) scheme, and distributes encrypted documents to storage servers. The servers may include a mix of third-party cloud storage and local in-house storage servers.

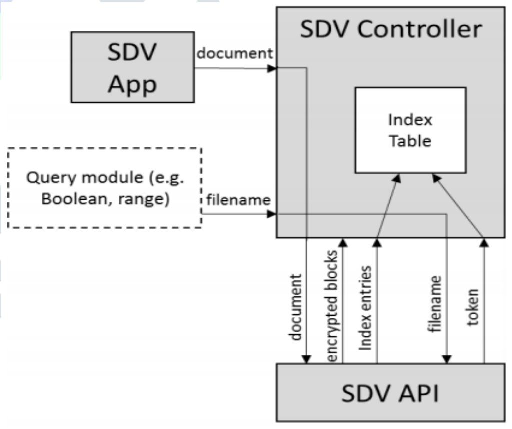

Figure 2. SDV controller—indexing and search

SDV controller—indexing and search: The SDV app submits a document to the SDV controller, which forwards the document to the SDV API. The SDV API then executes the underlying SSE scheme. The document is divided into blocks. The encrypted blocks, together with an index entry of the document are returned to the SDV controller, which then inserts the index entry to a query index table, and forwards the encrypted blocks to their respective servers based on the index entry. The index entry contains a token generated based on filename, and also information on where to store/retrieve the encrypted blocks of the document with the filename. The search then is simple: submit a filename to the SDV controller and a token is generated by the SDV API to search the index table. Given such a setting, a query module for more expressive search (e.g., Boolean) can be added as an additional module, as long as the search result for this module is a list of filenames.

II.RELATED WORK

Searchable encryption schemes enable the clients to store the encrypted data to the cloud and execute keyword search over ciphertext domain. Due to different cryptography primitives, searchable encryption schemes can be constructed using public key based cryptography or symmetric key based cryptography. The first symmetric searchable encryption (SSE) scheme, and the search time of their scheme is linear to the size of the data collection. 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. 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, multi-keyword boolean search ranked search, and multi-keyword ranked search etc. Multi-keyword boolean search allows the users to input multiple query keywords to request suitable documents. Among these works, conjunctive keyword search schemes only return the documents that contain all of the query keywords. Disjunctive keyword search schemes return all of the documents that contain a subset of the query keywords. Predicate search scheme are proposed to support both conjunctive and disjunctive search. All these multikeyword 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 work shave realized the ranked search using order-preserving techniques, but they are designed only for single keyword search. 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 according 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. 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 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 support the insertion and deletion of the documents. There are also several dynamic searchable encryption schemes. In the work of the each document is considered as a sequence of fixed length words, and is individually indexed. This scheme supports straightforward update operations but with low efficiency. Goh 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 constructed an encrypted inverted index that can handle dynamic data efficiently. But, this scheme is very complex to implement. Subsequently, as an improvement, proposed a new search scheme based on tree-based index, which can handle dynamic update on document data stored in leaf nodes. However, their scheme is designed only for single keyword Boolean search. In Cash et al. presented a data structure for keyword/identity tuple named "TSet". 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. MULTI-KEYWORD RANKED SEARCH OVER ENCRYPTED (MRSE)

Now a day's cloud computing has become essential for many utilities, where cloud customers can slightly store their data into the cloud so as to benefit from on-demand high-quality request and services from a shared pool of configurable computing resources. Its huge suppleness and financial savings are attracting both persons and enterprise to outsource their local complex data management system into the cloud. To safe guard data privacy and struggle unwanted accesses in the cloud and away from, sensitive data, for example, emails, personal health records, photo albums, videos, land documents, financial transactions, and so on, may have to be encrypted by data holder before outsourcing to the business public cloud; on the other hand, obsoletes the traditional data use service based on plaintext keyword search. The insignificant solution of downloading all the information and decrypting nearby is clearly impossible, due to the enormous amount of bandwidth cost in cloud scale systems. Furthermore, apart from eradicating the local storage management, storing data into the cloud supplies no purpose except they can be simply searched and operated. Thus, discovering privacy preserving and effective search service over encrypted cloud data is one of the supreme importance.

In view of the potentially large number of on-demand data users and vast amount of outsourced data documents in the cloud, this difficulty is mostly demanding as it is really difficult to gather the requirements of performance, system usability, and scalability. On the one hand, to congregate the efficient data retrieval requirement, the huge amount of documents orders the cloud server to achieve result relevance ranking, as an alternative of returning undifferentiated results. Such ranked search system allows data users to discover the most appropriate information quickly, rather than burdensomely sorting during every match in the content group. Ranked search can also gracefully remove redundant network traffic by transferring the most relevant data, which is highly attractive in the "pay-as-you-use" cloud concept. For privacy protection, such ranking operation on the other hand, should not reveal any keyword to related information. To get better the search result exactness as well as to improve the user searching experience, it is also essential for such ranking system to support multiple keywords search, as single keyword search often give up far too common results. As a regular practice specifies by today's web search engines i,e Google search, data users may lean to offer a set of keywords as an alternative of only one as the indicator of their search interest to retrieve the most relevant data. And each keyword in the search demand is able to help narrow down the search result further. "Coordinate matching", as many matches as possible, is an efficient resemblance measure among such multi-keyword semantics to refine the result significance, and has been widely used in the plaintext information retrieval (IR) community. Though, the nature of applying encrypted cloud data search system remains a very demanding task in providing security and maintaining privacy, like the data privacy, the index privacy, the keyword privacy, and many others.

Encryption is a helpful method that treats encrypted data as documents and allows a user to securely search through a single keyword and get back documents of interest. On the other hand, direct application of these approaches to the secure large scale cloud data utilization system would not be necessarily suitable, as they are developed as crypto primitives and cannot put up such high service-level needs like system usability, user searching experience, and easy information discovery. Even though some modern plans have been proposed to carry Boolean keyword search as an effort to improve the search flexibility, they are still not sufficient to provide users with satisfactory result ranking functionality. The solution for this problem is to secure ranked search over encrypted data but only for queries consisting of a single keyword. The challenging issue here is how to propose an efficient encrypted data search method that supports multi-keyword semantics without privacy violation. In this paper, we describe and solve the problem of multi-keyword ranked search over encrypted cloud data (MRSE) while preserving exact system wise privacy in the cloud computing concept. Along with various multi-keyword semantics, select the efficient resemblance measure of "coordinate matching," it means that as various matches as possible, to confine the significance of data documents to the search query. Particularly, inner product similarity the numbers of query keywords show in a document, to quantitatively calculate such similarity assess of that document to the search query.

For the period of the index construction, each document is associated with a binary vector as a sub-index where each bit signifies whether matching keyword is contained in the document. The search query is also illustrates as a binary vector where each bit means whether corresponding keyword appears in this search request, so the resemblance could be exactly calculated by the inner product of the query vector. With the data vector. On the other hand, directly outsourcing the data vector or the query vector will go against the index privacy or the search privacy. To face the challenge of cooperating such multi keyword semantic without privacy breaches, we propose a basic idea for the MRSE using secure inner product computation, which is modified from a secure k-nearest neighbor (kNN) method, and then give two considerably improved MRSE method in a step-by-step way to accomplish different severe privacy needs in two risk models with enlarged attack competence.

IV. DISCUSSION

In this section, we discuss other properties of our scheme. Eliminating Random Oracle Assumption. Our construction is ready to be deployed in the standard model with small modifications: The hash function is replaced by a proper PRF. Then, instead of sending the key and asking the server to compute all respective hash values for search and deletion operations, the client computes and sends all PRF outputs to the server, similar to file addition. The server, given the required PRF outputs, decrypts the requested cells and acts according to the requested operation. Our construction in the standard model inherits and preserves all properties of the random oracle model counterpart. To the best of our knowledge, this is the most efficient SSE construction in the standard model with forward privacy. Another important advantage of this construction is that the server is not expected to even evaluate hash functions anymore. It only XORs the received values with those in the specified cells to extract the identifiers of files that constitute the answer. This means the server is no longer performing even simple cryptographic operations, and renders our scheme to be deployable in almost all existing cloud environments.

Regarding efficiency, the client's computation and token size for search is increased from O(1) to O(d). The server computation is still O(d), without any hash function evaluations. File insertion continues working in the same manner, i.e., with O(r) client and server computation, and communication. If supported, deletion asymtotics will be increased to O(r), similar to search.

Parallelism. Our scheme is ready to benefit from parallelism. It evaluates O(d) and O(r) hash functions for search and update, respectively. Each hash function evaluation is independent, and takes the respective key k and a sequence number s as input: h(k,s). Therefore, hash function evaluations can all be done in parallel. This allows the service provider to distribute the load on p available processors, achieving O(d/p) and O(r/p) search and update costs, respectively. The most efficient known schemes support search with O(d) [6, 20] or O((d logn)/p) [19] cost and update with O(r) [6, 20] or O((mlogn)/p) [19] cost. Further replication, distribution, and load balancing mechanisms can be employed to improve performance and data availability [2, 28, 30].

Batch update. Our scheme in its current state supports batch update (i.e., file insertion). In contrast to the schemes that store index data in a 'sequential' way [6, 20, 26, 29], our scheme stores index data, even for the same file or keyword, in random locations. Therefore, the client can send the updates corresponding to a number of files, without any order, and ask the server to add them all into the index. This also helps to reduce the leakage of adding files one-by-one, since the server does not learn how many keywords each file contains.

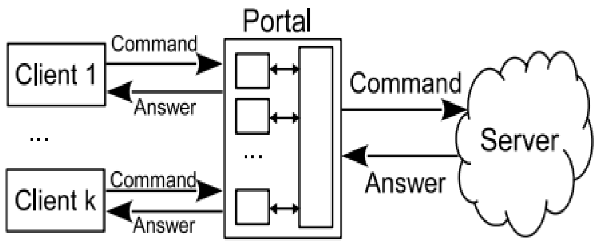

Fig. 3. The portal serves all clients

Organizational portal.. When the number of (keyword, file ID) mappings increases, our scheme requires a larger storage at the client side (if it is not outsourced). In an organization outsourcing a huge number of files, it is not reasonable to replicate the same set of local information over all the clients. Using a portal server solves the problem. It is a local (and hence a trusted) entity that stores the same information as a regular client, and serves all clients of the organization. The portal receives the clients' requests, prepares them according to the scheme in use, and sends the resultant command to the server. On receipt the server's answer, relays it to the respective client. In addition, using the local buffer on the portal improve performance of the whole scheme.

V. CONCLUSION

Ensuring forward privacy is an important step to mitigating attacks on SSE. We propose a dynamic SSE scheme that provides forward privacy with better performance than any previous scheme and without needing any asymmetric operations. Our scheme reduces the required server computation, and limits the server role to mostly storage rather than computation. Hence, our scheme can be employed by a broader range of service providers. In this work, we have developed and implemented the SDV system prototype that runs on a multi-server SSE scheme, and showed that encrypting documents into blocks and uploading to several cloud storage providers entails acceptable cost overhead compared to uploading original files unencrypted. Our system is easy to use and robust, with an intuitive web interface that is compatible with any browser. It solves the problem of privacy and searchability of encrypted documents with minimal leakage as no single cloud service provider will have the complete document block set.

REFERENCES

[1] I. H. Akin and B. Sunar, "On the difficulty of securing web applications using cryptdb," in IEEE International Conference on Big Data and Cloud Computing Conference, 2014.

[2] R. Alonso-Calvo, J. Crespo, M. Garc'ia-Remesal, A. Anguita, and V. Maojo, "On distributing load in cloud computing: A real application for very-large image datasets," Procedia Computer Science, vol. 1, no. 1, 2010.

[3] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in EUROCRYPT'04, 2004.

[4] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in Theory of Cryptography, 2007.

[5] C. Bösch, A. Peter, B. Leenders, H. W. Lim, Q. Tang, H. Wang, P. Hartel, and W. Jonker, "Distributed searchable symmetric encryption," in IEEE Conference on Privacy, Security and Trust (PST), 2014.

[6] R. Bost, "σοφος: Forward secure searchable encryption," in ACM Conference on Computer and Communications Security (CCS), 2016.

[7] D. Cash, P. Grubbs, J. Perry, and T. Ristenpart, "Leakageabuse attacks against searchable encryption," in ACM Conference on Computer and Communications Security (CCS), 2015.

[8] D. Cash, J. Jaeger, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Rosu, and M. Steiner, "Dynamic searchable encryption in very-large databases: Data structures and implementation," in Network and Distributed System Security (NDSS), 2014.

[9] D. Cash, S. Jarecki, C. Jutla, H. Krawczyk,

[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.M.-C. Roşu, and M. Steiner, "Highly-scalable searchable symmetric encryption with support for boolean queries," in CRYPTO'13, 2013.
[14] B. Wang, S. Yu, W. Lou, and Y. T. Hou, "Privacy-preserving multikeyword fuzzy search over encrypted data in the cloud," in IEEE INFOCOM, 2014.
[15] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in Applied Cryptography and Network Security. Springer, 2004, pp. 31–45.