INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X

Impact factor: 4.295

(Volume 4, Issue 2)

Available online at: www.ijariit.com

Privacy- Preserving Multi Keyword Search in Cloud Computing

Mansour Alshaikhsaleh
mansouralshaikh@oakland.edu
Oakland University, Rochester, Michigan, USA

Prof. Mohamed Zohdy
zohdyma@oakland.edu
Oakland University, Rochester, Michigan, USA

ABSTRACT

The current study is intended to present an in-depth analysis of literature relative to privacy-preserving multi-keyword search over encrypted cloud data. The multi-keyword ranked search that the study proposes has the capacity to post the exact matching documents only. This paper essentially seeks to fill the perceived research gaps from previous studies in multi-keyword search in cloud computing. To be precise, the current paper considers three peer-reviewed research articles concerning multi-keyword search in cloud computing. As such, it analyses the methods, approaches, and techniques that have been proposed in each of the three studies while presenting viewing these aspects in a critical limelight. The central problem focused on each of the three selected articles is keyword privacy. This draws from the fact that despite the capacity of perceived approaches to generate cryptic trapdoors which conceal keywords, there is a possibility that cloud service providers through statistical analysis can estimate the keywords. The second concern that has been discussed in these research articles is trapdoor un-linkability. This arises mostly from deterministic trapdoor generation functions which give the cloud service provider an easy avenue of accumulating keyword frequency. Lastly, surveyed studies also reveal concerns relating to access patterns whereby although a many of encryption techniques can be deployed to conceal the access pattern, there remain efficiency concerns with such encryption.

Keywords: *Privacy, Preserving Search, Multi-keyword, Cloud Computing.*

1. INTRODUCTION

As a result of rapid data expansion owing to the needs of the modern organization, and partly fueled by the information age's obsession with data, most organizations, as well as individuals, reduce the cost of data maintenance by seeking external alternatives – the cloud. The problem that comes with cloud solutions is that the cloud server and the user are usually in different domains thus raising the question of trust. More importantly is the fact that there is a need to consider the risk that the user exposes themselves to by allowing third-party storage partners access to their data. This is the case owing to the fact that providers of cloud services enforce user data security through virtualization as well as mechanisms such as firewalls. The problem is that such approaches do not protect the users from the providers themselves. This draws from the fact that the provider has full control of the server's hardware not to mention other lower software stack levels. This means that in the event of a curious employee or an ethical provider with intent for easy profiteering at the cloud service provider, they may use deploy the information for unauthorized purposes. An easy solution that technology offers is encryption which allows the user to protect sensitive data and at the same time also combat unsolicited access. However, there arises a problem in that conventional model approaches are not applicable to cloud data. Conventional informational retrieval (IR) gives the user multi-keyword ranked search. This is the same capability that cloud servers need to extend to the user. The question that remains relates to how cloud servers can generate the same solution while also protecting information and preserving the privacy of the user.

2. BACKGROUND

The encryption techniques that are described in existing studies. Secondary research studies cite the development of a searchable index with the capacity to store a list of mapping from keywords to the opposite set of documents that contained the word [5]. In this sense when a user inputs a keyword, the system generated a trapdoor for the corresponding keyword which was the subsequently submitted to the server in the cloud [5]. When the server in the cloud received the trapdoor, it initiated a comparison between the index and trapdoor before ultimately returning all the documents that contained the keyword that the user submitted in their query. This study builds on research that addresses the problem of secure ranked searches in the cloud.

© 2018, www.IJARIIT.com All Rights Reserved

3. PROBLEM FORMULATION

Despite its widespread adoption and use, cloud computing remains to be challenged by rising privacy concerns. With a constantly mounting volume of data being stored, sent and shared over the cloud, the risk of privacy breaches is also increasing steadily. When one considers cloud service providers as a cost-cutting alternative to in-house data maintenance, there are three entities that come to mind – the data user, the data owner, and the cloud server. The owner of the data is the entity that owns a series of documents labeled $\mathcal{F}$ which they seek to outsource to the cloud service provider after encrypting the data in the form $\mathcal{C}$. However, in order to make it possible for such data stored in the cloud in the form $\mathcal{C}$ to be accessed by the user, the owner will begin by developing an encryption index $\mathcal{I}$ which corresponds to $\mathcal{F}$. The versions that are sent to the cloud service provider as such include the encrypted search index $\mathcal{F}$ and the encrypted version of the documents in the form of $\mathcal{C}$. When the user thus searches the document collection for t given keywords, they acquire a corresponding trapdoor T through control mechanisms of the query such as broadcast encryption [2]. When the cloud service provider receives T, they direct it to the index I before ultimately returning the index documents that corresponds to the query. In order to have an efficient and accurate document retrieval exercise, the cloud service provider must rank the results of the search using some form of a ranking criteria [4]. Additionally, in order to cut down the cost of communication, the data user must be allowed the capacity to send an optional number together with the trapdoor T which in essence means that the server only returns top-k files relevant to the initiated query.

Although the study perceives the cloud server as an honest host, it nevertheless presumes it to be increasingly curious. This point of view is consistent with past studies.

4. CRITICAL ANALYSIS OF APPROACHES, METHODS AND TECHNIQUES

4.1 The Minhash Function

The proposed scheme laid down by Orencic et al search in an encrypted cloud uses an encrypted searchable index which the data owner generates before outsourcing to the cloud service provider. When a user queries the cloud server, the server proceeds to with a comparison with the searchable index before posting results without acquiring additional information thus fulfilling the efficiency concerns (Orencic et al). In order to critically evaluate the study by Orencic et al, it is primarily important to build a basic understanding of their proposed solution.

The authors employ the bucketization approach in which data is partitioned into segments and each object of data is distributed into numerous buckets using the minhash functions. In minhashing each of the owner's documents is represented by a small set referred to as a signature. With the signature technique, the most critical characteristic is its capacity to compare two signatures before estimating the distance between any two underlying sets without availing additional data during the query process. Despite the fact that the precise similarity is hard to deduce from the signatures, the technique is ideal and can be relied upon to approximate within a very small margin of error. At the same time, the accuracy of the similarity also grows with larger signatures. The composition of the signatures is diverse and include several elements with each element developed using minhash functions. With a finite set of elements Δ, where the permutation on Δ is denoted, has P9i) as the i^{th} element in the permutation of P (Orencic et al). Therefore the definition of Minhash of any given set $D \subseteq \Delta$ under permutation P is expressed as follows:

$$hP(D) = \min(\{i \mid 1 \leq i \leq |\Delta| \wedge P[i] \in D\}). \text{ (Orencic et al).}$$

Subsequently, for each signature, Orencic et al contend that different random permutations on Δ are employed deriving the final signature of any given set D as follows:

$$Sig(D) = \{hP1(D), \ldots, hP\lambda(D)\}, \text{ (Orencic et al).}$$

The bucket-id is employed to identify each object in the bucket. The working of this technique is such that in mapping the objects, the number of buckets with two coinciding objects rises with an increase in their similarity (Orencic et al). This is to say that while two similar objects coincide in all buckets, the number of common buckets reduces with object dissimilarity (Orencic et al). The data owner derives the secure index that the authors propose with the use of three specific phases including extraction of features, construction of bucket index, and encryption of bucket index (Orencic et al)

In constructing the bucket index, Orencic et al begin with developing a minhash structure. This is done through selection of λ random permutations done on a set of as many search terms as can be conceived (Δ). The researchers then apply the minhash to the first values of every feature set $F*i = \{wi1, \ldots, wiz\}$ before deriving a signature for each document as follows:

$$Sig(Di) = \{hP1(F*i), \ldots, hP\lambda(F*i)\}. \text{ (Orencic et al)}$$

Significant to note is the fact that each signature element of a document is defined as a keyword for the same document. In this approach, Orencic et al map each document to λ buckets using the documents. If $hPi(F*j) = wk$, the authors create a bucket with Bik as the bucket identifier together with relevancy scores and identifiers of the documents that fulfill this property which is added to the bucket (Orencic et al). The content of the bucket is also definitive – the vector of integer elements of l which denotes the number of documents in the outsourced dataset (Orencic et al). As such, if VBik I the integer vector and Bik is the bucket identifier Orencic et al note that $VBi[id(Dj)] = rsjk$ remains true if and only if $hPi(F*j) = Bik$, otherwise, the authors note that, $VBi[id(Dj)] = 0$ (Orencic et al).

The search is secure because given a query Q for instance, the server determines the encrypted vectors (VBjk) that correspond to the bucket identifiers. The encrypted identifiers $EV = \{V1, \ldots, V\lambda\}$ are then sent back to the user. The user decrypts the vectors

© 2018, www.IJARIIT.com All Rights Reserved

before ranking the data identifiers (Orencic et al). Nonetheless, this process of bucket identification and vector decryption interprets into a greater time required for processing. This resultantly questions the claim of efficiency.

To begin with, the Minhash approach proposed by Orencic et al can be considered somewhat biased in a sense that the authors contradict their own opinions about this method. To be precise, the authors state in the beginning of the paper that Minhash serves as an efficient privacy preserving method; however, the authors follow this contention by a contradictory statement that Minhash assists in improving effectiveness. This makes it somewhat ambiguous in determining whether the Minhash function instils efficiency, effectiveness or both at the same time? Further analysis of the study by Orencic et al highlights the privacy-preserving capability of their proposed approach in addition to being efficient and effective. However, these capabilities have only been proven by the authors under controlled and well-executed experiments while in reality, scenarios may differ significantly thereby reflecting on the actual performance of the Minhash function approach. It is imperative to consider that Minhash is a probabilistic approach that is relatively new. Thus, concluding their proposed method as privacy-preserving, efficient and effective may be before time and there is a need for a greater number of studies to test and verify the claimed characteristic attributes therein.

4.2 Verifiable Privacy

In their study, Sun et al [4] addressed some of the setbacks related with developing efficient and practical encrypted data search functionalities with the capacity to support multi-keyword queries, result verification and result ranking [4]. In order to achieve multi-keyword ranked data search, the authors adopted the cosine measure in evaluating similarity scores [4]. Particularly, the study divided the lengthy original document index vector Dd into several sub-vectors in a way that each sub-vector Dd; I stood for a subset of keywords Ti of T, and thus became part of the i^{th} level of the I index tree. In this approach, the query vector Q is divided in a similar manner as D_d. Letting Qi be the same query vector at the i^{th} level as the authors explain the final similarity score for the document in the server is derived by adding all the scores of from all levels. On the basis of the derived similarities the cloud service provider determines document relevance to the user. At the same time, because the technique the level wise secure inner product platform, the query vector together with the document index vector is concealed. The work by Sun et al [4] is worth consideration particularly because of the fact that it builds on the previous research studies. For instance, prior to presenting experiment results, it presents a thorough account of searchable encryption with a single keyword, verifiable search based on authenticated index structure, and searchable encryption with multiple keywords [4]. The researchers have adopted the cosine similarity measure, incorporating TF x IDF weight, which according to the researchers enables in achieving more search accuracy [4]. Furthermore, the researchers have divided the long vector index into multiple layers, while proposing a tree-based index structure [4]. As for the requirement of search efficiency, the researchers have adopted the MD-algorithm [4].

As such, the cosine model adopted by the researchers divides long vector D_d into multiple sub-vectors such that each subvector $D_{d;i}$ represents a subset of keywords T i of T , and becomes a part of the *ith* level of the index tree I [4]. Their verifiable privacy solution is based on three key search privacy requirements, including index confidentiality, query confidentiality, query unlinkability and keyword privacy [4]. GenIndex *DC,SKi* , GenQuery, and SimEvaluation are core aspects of the tree-based search algorithm proposed in their study [4].

Throughout the research article compiled by Sun et al [4], the use of the index is evident. While the authors claim their proposed framework to be efficient and effective due to the multi-keyword ranked search capability, one can never ignore the cons associated with using the index, i.e. using index accompanies significant updating and storage overhead costs. This points to a contrary direction than what is claimed by the authors. As such, a framework using index appears to be more appropriate for read-only data collections. This is because reducing overhead costs is among the most pertinent motives behind utilizing the Cloud. Hence, a company using the cloud as a less-expensive alternative would not be able to benefit from the approach proposed by Sun et al [4] without paying for additional overheads. Besides, utilizing the proposed framework on read-only data collections would be pointless.

4.3 Privacy Perserving Multi-Key Word Search

In their study, Cao et al [1] sought to achieve three main goals. Firstly, the authors sought a tool for multi-keyword ranked querying whereby they sought to develop search schemes that would allow multi-keyword query and at the same time also provide result similarity ranking in order to retrieve data effectively as opposed to posting results that were undifferentiated [1]. Secondly, the authors also sought to preserve the privacy of a user's queries by preventing the cloud service provider from accessing additional information from the dataset and the index generated during the query. The third goal related to efficiency. Here Cao et al's goal were to have a system with minimal communication as well as computational overheads [1]. The authors thus propose a multi-keyword ranked search over encrypted cloud data (MRSE) approach that preserves system-wise privacy. From the numerous multi-keyword semantics that is available, the study opted for coordinate matching – a similarity measure [1]. In coordinate matching, the query uses as many matches as can be derived to capture relevant data related to the user's query. Particularly, the study employs inner product similarity for quantitative evaluation of the search query to the document. In index construction, the technique relates each binary vector to a document as a sub-index.

The MRSE comprises four algorithms. The first one is that Setup (1^{ℓ}) whereby the security parameter being ℓ the data owner symmetric key output is denoted as SK. The second algorithm, BuildIndex denoted as F, SK, has the data owner, based on the data set F, developing a searchable index I which they encrypt by the symmetric SK before outsourcing the developed index to the cloud. Subsequent to index development, the collection of files are encrypted autonomously before the owner outsources them to the cloud [1]. The third algorithm, TrapDoor (*W*) operates on the basis that with t amount of keywords in the TrapDoor *W* as the primary input, the algorithm generates TrapDoor Tw as correspondence. The last algorithm is Query expressed as Tw k, I. With the query, when the cloud server receives a query request expressed as Tw k, it proceeds with a ranked search through the existing index I using the trapdoor Tw as its main backdrop of operation. The result is Fw which ideally corresponds to the ranked identification list of all top k known documents sorted according to their similarity with the trapdoor W [1]. According to the authors both access

© 2018, www.IJARIIT.com All Rights Reserved

control and search, control is not within the scope of their study. This gap in the study is what motivates the current research because it relates to managing the users' access to outsourced data.

In relation to privacy requirements for MRSE it relates to searchable encryption whereby the server gains no additional knowledge other than the search results. There are a number of privacy requirements that are related to the MRSE model. Firstly, Cao et al [1] point out that the data owner can invoke the conventional symmetric key cryptography in encrypting the information prior to outsourcing it thereby preventing the cloud service provider from having unauthorised access to the private information. With index privacy on the other hand, in the event the cloud service provider makes inferences relating to keywords and encrypted documents, there is a high likelihood that they can use such inferences to decipher a major document thus pilfer information from the server for their own use. There is a need for the searchable index to be developed in such a way that the cloud service provider does not have any chance of association attacks. Although both index and data privacy are both guarantees that come by default, there are a number of search privacy requirements that characterize the process of a search query that is inherently complex and more difficult to resolve.

The study by Cao et al [1] appears to have limitations with regards to the privacy requirements that it establishes. For instance, their research states the ability of a data owner to encrypt data through traditional symmetric cryptography, which will prevent the cloud server from prying into the data that will be outsourced [1]. If future researchers were to consider this very specific statement, their work in privacy-preserving multi-keyword ranked search would require a whole new beginning. If a data owner has to utilize cryptographic encryption, then what are the proposed solutions worth? Undoubtedly, the proposed framework should itself be capable of preserving privacy especially if the researchers claim it to be potent. Secondly, similar to the model proposed by Sun et al [4], Cao et al [1] also use the index in their solution, while claiming their solution to be cost-effective. The current experiments and results shared in their study fail to substantiate how their solution saves overheads while improving efficiency and effectiveness.

5. ESTABLISHED RESEARCH GAPS RELATED TO PRIVACY

One such difficulty arises with key word privacy. This is attributed to the tendency of users to prefer concealing their search from others including the cloud service provider whereby the most pressing concern is masked what they are searching. These usually constitutes the keywords that correspond to the trapdoor. The problem is that despite the fact that the trapdoor can be generated in a cryptographic way for the protection of the keywords of a search, there is a possibility that the cloud service provider, through statistical analysis of the search result, can come up with almost if not accurate estimates [1].

The second privacy concern in multi-keyword searches relates to trapdoor un-linkability. This concern draws from the fact that the trapdoor generation tool, as opposed to being randomized is deterministic. This is important because it prevents the relationships between trapdoor from being revealed to the cloud service provider. With a deterministic trapdoor generation, the cloud service provider is handed the advantage of accumulating frequencies of different queries that regard different keywords [1]. This, in essence, means that the privacy requirement of the keyword is undermined and thus it leads to privacy concerns.

Thirdly, there is also a concern relating to access. The pattern of access refers to the sequence of search results within the ranked data, where every search result relates to a set of documents in a rank order [1]. If a query is linked to a set W, whereby its search result is expressed as FW and consists of the identification list of all documents by their relevance, the resulting access pattern can be generated in sequence. Despite a number of proposed encryption models in which the main framework is based on private information retrieval, the analyzed studies are not designed to conceal the pattern of access.

In view of the aforementioned limitations and issues, a starting point would be to initially sort out the privacy-based requirements that align correctly with an innovative and efficient system. As such, the privacy requirements should at its very least, not require users to encrypt their data/keywords through other encryption methods while making a multi-keyword search. Additionally, a more appropriate model for privacy-preserving multi-keyword search could comprise of the following four modules as adopted from the study by Dhumal & Jhadav [3];

- Binary data generation.
- Data ciphering.
- Data user access control.
- Data user query.

Adherence to these four modules could help ensure the accuracy and effectiveness of multi-keyword searches via the cloud. In addition, the requirement of efficiency could be ensured by the use of the BlowFish algorithm that is specifically meant for decryption and encryption of data, where encryption comprises of 16 rounds with a 64 bit data element input 'x', divided into two equal halves of 32 bits each, i.e. xL, xR. Then, for i=1 to 16, xL=xL XOR Pi and xR=F(xL)XOR xR [3]. Once the 16th round is completed, xL is swapped with xR. It is anticipated that through adoption of this specific model, which is also relatively newer than the rest discussed in this paper, a more efficient and effective multi-keyword search method could be devised that is also accurate in terms of privacy preservation.

6. CONCLUSION

The current paper defines multi-keyword ranked queries conducted for data stored in the cloud, while focusing on privacy preservation as the central problem. There are a number of privacy concerns that arise. The first concern relates to key word privacy which develops from the fact that, despite the ability to generated cryptic trapdoor, there is an increasing likelihood of a cloud service provide, through statistical analysis, an accurately estimate keywords. The second concern that is pointed out relates to trapdoor unlinkability where a deterministic trapdoor generation function allows the cloud service provider some leeway in

© 2018, www.IJARIIT.com All Rights Reserved

accumulating search request frequencies. The third concern has to with the access pattern. These concerns present rich areas for further inquiry. In the current study, 3 research articles from peer-reviewed journals were critically analysed in-depth. Results of the analysis clearly show a range of limitations and the need for more extensive research in a highly structured manner

7. REFERENCES

[1] Cao, Ning, et al. "Privacy-preserving multi-keyword ranked search over encrypted cloud Data." IEEE Transactions on parallel and distributed systems 25.1 (2014): 222-233.
[2] Chen, Li, et al. "An Efficient and Privacy-Preserving Semantic Multi-Keyword Ranked Search over Encrypted Cloud Data", International Journal of Security and its Applications, 8(2) 323-332 (2014).
[3] Dhumal, Amol and Jadhav, Sanjay. "Confidentiality-Conserving Multi-Keyword Ranked Search above Encrypted Cloud", Procedia Computer Science, 79, 845-851 (2016).
[4] Sun, Wenhai, et al. "Verifiable privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking" IEEE Transactions on Parallel and Distributed Systems 25.11 (2014): 3025-3035.
[5] Zhangjie, Fu, et al. "Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing." IEICE Transactions on Communications 98.1 (2015): 190-200

© 2018, www.IJARIIT.com All Rights Reserved