ISSN No. 0976-5697

Volume 6, No. 3, May 2015 (Special Issue)

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A Comparative Study of Index-Based Searchable Encryption Techniques

M. Samuel John
Department of Computer Applications,
V.R.Siddhartha Engineering College,
Vijayawada, INDIA.
E-mail: msjohn@vrsiddhartha.ac.in

P. SumaLatha
PG Student, Dept. of Computer Applications,
V.R.Siddhartha Engineering College,
Vijayawada, INDIA
E-mail:suma.perumalla@gmail.com

M. Joshuva
Department of Computer Science and Engineering,
Sri Prakash College of Technology, Rajahmundry,
East Godavari, A.P., INDIA
E-mail: m.jashua@gmail.com

Abstract: Now a days, many organizations as well as individuals prefer to store their data in cloud storage servers. Storing sensitive data in untrusted cloud storage server always leads to concerns about its confidentiality. In order to provide data confidentiality, encryption technique is used. To protect the sensitive information like emails and health records and, to reduce the security risks, cloud tenants encrypt the information before storing the date in the cloud. Encryption scrambles the plaintext data. But, searching for a keyword in an encrypted file is not as easy as searching in a plain text file. So, over the years many searchable encryption schemes have been proposed. Searchable encryption enables users to perform search on encrypted data. In this paper we compare the performance of index based searchable encryption schemes.

Keywords: Cloud Security, Confidentiality, Index based Searchable Encryption, Searchable Encryption, Searchable Symmetric Encryption.

I. INTRODUCTION

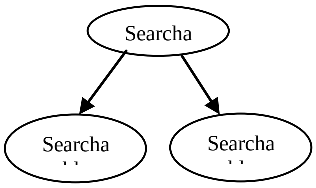
Although cloud storage servers offer many advantages, the security of our personal data is a big concern. Even though encryption provides security, it complicates the search operation on cipher text. One solution for searching is to download all the documents, decrypt them and start searching. But it is not possible to download all the documents every time whenever search is needed. The other solution is searching in the encrypted data itself.

One searching technique is to encrypt the keyword and sequentially scan all the documents for that particular keyword. Even this method suffers from a disadvantage. Sequential Search is inefficient if large amounts of data are stored in the cloud storage provider. The other technique is to create an index file, which consists of keywords and pointers to files in which the keywords appear, and start searching the index file for the keyword. This is called index-based searching.

Organization of the paper: The remainder of the paper is organized as follows: Section 2 presents an outline of search on encrypted data, Section 3 describes index based searchable encryption. We show the experimental study and results in section 4 and conclude the paper in section 5.

II.SEARCH ON ENCRYPTED DATA

There are different approaches to search on encrypted data. These approaches are based on property preserving encryption[1], functional encryption[2], fully homomorphic encryption[3], oblivious RAMs[4], secure two party communications[5], ranked keyword search[6], and searchable encryption [7].



Searchable encryption can be divided into two distinct types based on the number of keys involved in the encryption. They are, searchable symmetric encryption (SSE) and searchable asymmetric encryption (SAE).

Searchable Symmetric Encryption:

In Searchable symmetric encryption, either the entire file is encrypted or each word in a file is encrypted using a symmetric encryption algorithm like DES or AES. Since the key is with the owner of the file, he can encrypt the word for which he is searching and can easily search in the encrypted file. The first searchable symmetric encryption scheme is proposed by Song, Wagner and Perrig [7]. Later many static as well as dynamic searchable symmetric encryption techniques are proposed.[8-10].

Searchable Asymmetric Encryption:

The first ever searchable asymmetric encryption technique is proposed by Boneh *et al* [11]. Hacigumus[12] proposed a scheme to support range queries on encrypted data.

III. INDEX BASED SEARCHABLE ENCRYPTION

The Index consists of a set of keywords and document identifiers or pointers to the documents in which the keyword is present. It is used to accelerate the search process. Instead of

CONFERENCE PAPER
4th National Conference on Recent Trends in Information Technology 2015 on 25/03/2015
Organized by Dept. of IT, Prasad V. Potluri Siddhartha Institute of Technology, Kanuru, Vijayawada-7 (A.P.) India

searching all documents sequentially for a keyword, one can first search in the index file. If the keyword is present in any document(s), the document identifier(s) is returned.

The owner of the documents encrypts all the documents and creates an index file and encrypts it. While encrypting the index file the owner has two options. One is to encrypt keywords and leave the corresponding document identifiers in the clear and the other way is to encrypt both keywords and document identifiers. The disadvantage of the first method is, it leaks some information which can be used by an attacker for statistical attacks.

The drawback of the second method is, it requires extra time to decrypt the results to retrieve the documents.

There are two methods to construct indexes.

1.forward index

2.inverted index

A forward index is a mapping from documents to the keywords.

Table I. Forward Index

Documents	***Keywords***
Doc1	Cryptography, AES, Digital Signatures
Doc2	RSA,AES,PKI
Doc3	RSA,DSS,PKI

An inverted index is a mapping from keywords to the corresponding documents.

Table II. Inverted Index

Keywords	***Documents***
Cryptography	Doc1
AES	Doc1, Doc2
Digital Signatures	Doc1
RSA	Doc2, Doc3
DSS	Doc3
PKI	Doc2, Doc3

In a text book, a forward index is the ‘table of contents’ placed at the beginning of the book and an inverted index is the ‘index’ at the end of the text book.

Several methods based on the inverted index have been proposed. Some of these techniques are proposed by Goh[13], Curtmola[14], Golle[15], and Tang[16]. One of the disadvantages of using index based searching is that the owner has to update the index file whenever the individual documents are updated.

IV. Experimental study and Results

In this section we provide the results of our comparative study. In our experimental study, we used a standard symmetric encryption algorithm AES to encrypt all our documents and the index file.

Say we want to search for all the documents which has the keyword ‘PKI’ in it. In the forward index, we need to scan each row to check for the word ‘PKI’. So , we need to search 3 times to find that it exists in Doc2 and Doc3. But, in the inverted index, we need to find the keyword ‘PKI’ and find out that it exists in Doc2 and Doc3. So, inverted index is more effective when searching is done in large amount of data.

Users upload documents and related keywords of those documents. Server encrypts these documents and stores them in the database. Server encrypts and updates the index file. Whenever user wants to search for a document, user enters a related keyword for the document. Server processes the keyword, returns a secret key to user. (User has to enter that secret key to retrieve the documents). Server searches for the secret key in the index file. If it is found then, resultant documents are returned to the user. This technique allows a third party search.

If a third person, other than owner of documents, enters a keyword for search, he will receive a temporary secret key. This key is based on the keyword but, the third person cannot get or derive any information from this keyword. This keyword changes for every new search. Even though the third person searches for the same keyword in different times, he will get different secret key.

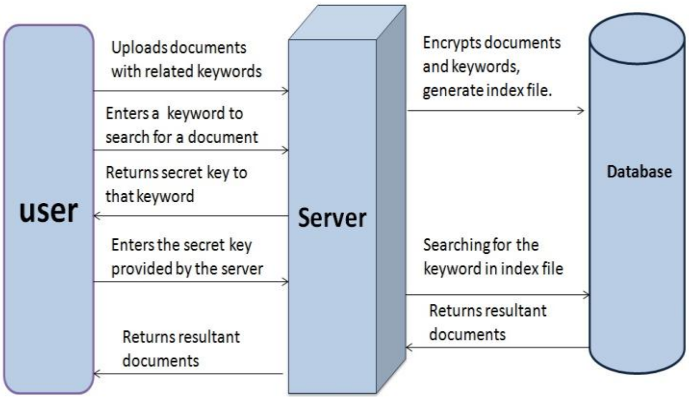

user
Uploads documents with related keywords
Enters a keyword to search for a document
Returns secret key to that keyword
Enters the secret key provided by the server
Returns resultant documents
Server
Encrypts documents and keywords, generate index file.
Searching for the keyword in index file
Returns resultant documents
Database

Figure 1. Workflow of searching

The following graph shows the performance of forward and inverted index based searchable encryption.

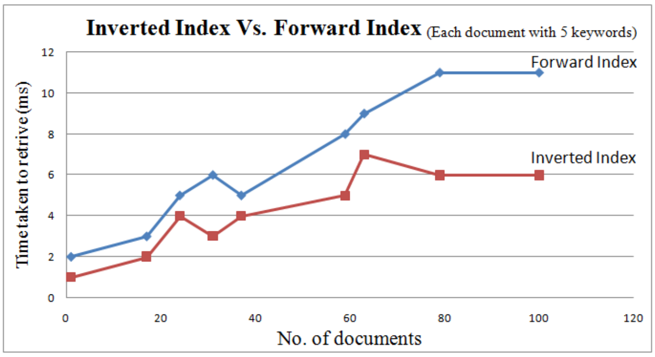


Figure 2. Forward Vs. Inverted Index (Each document with 5 keywords).

The following graph shows the performance of forward and inverted index searchable encryption when number of keywords per a document is increased.

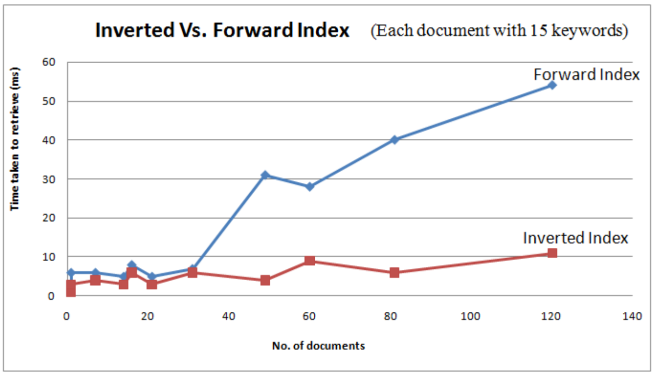


Figure 3. Forward Vs. Inverted Index (Each document with 15 keywords).

As we discussed earlier, forward index is made as number of keywords for a document. The searching takes a lot of time when the keyword is in the last document. But, if the keyword is present in the first document itself then, it takes less time. Whereas, inverted index results in equivalent search time, even it is in the first document or last document. Since inverted index is made as number of documents for a keyword.

The following graph shows the difference in search times in forward index file when the keyword to be searched is the first keyword and last keyword.

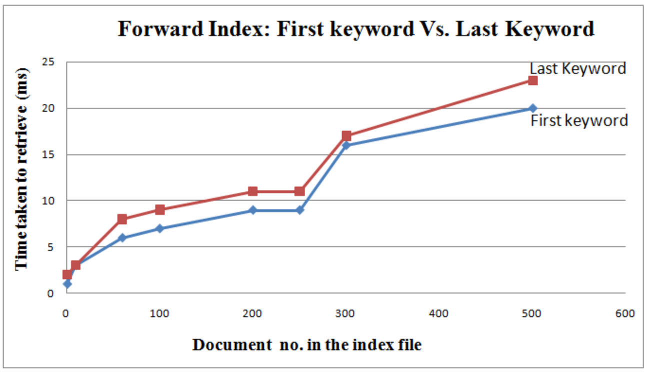


Figure 4. Forward Index: First keyword Vs. Last keyword.

To compare the performance of forward and inverted index based encryption techniques, we used Intel Core i3@2.20 GHz processor, Windows 7 Operating System and 4GB RAM.

V. CONCLUSION

In this paper, we illustrated the performance of forward index based and inverted index based searchable symmetric encryption techniques and provided practicality analysis. If a new file is added to the data store then, updating the forward index is simpler than updating the inverted index because, we just have to add a new line along with their key words. But, in inverted index, all the existing keywords are to be checked for their appearance in the new document and correspondingly their entries must be updated. Not only that, if changes are made frequently to a document then, index should be updated substantially. In this case forward index works better. It just adds one more line in the index file. But in the case of encrypted emails, which are not going to be modified later, inverted index is a better choice.

VI. REFERENCES

[1] M. Bellare, A. Boldyreva, and A. ONeill “Deterministic and efficient searchable encryption” in *Proc. of Crypto’07*, 2007, pp. 535–552

[2] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In *Advances in Cryptology - EURO-CRYPT'04*, volume 3027 of *Lecture Notes in Computer Science*, pages 506–522. Springer, 2004.

[3] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st annual ACM symposium on Theory of computing* , STOC ’09, pages 169–178, New York, NY, USA, 2009. ACM.

[4] O.Goldreich and R. Ostrovsky. “Software protection and simulation on oblivious RAMs”. *Journal of ACM*, 43(3):431–473, 1996

[5] A.C. Yao. “Protocols for secure computations” *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, pages 160-164, 1982.

[6] C. Wang, N. Cao, K. Ren, and W. Lou. “Enabling secure and efficient ranked keyword search over outsourced cloud data”. *IEEE Transaction on Parallel Distributed Systems* , 23(8):1467–1479, 2012

[7] D. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in *Proc. of the IEEE Symposium on Security and Privacy’00*, 2000, pp. 44–55.

[8] S. Kamara and C. Papamanthou. Parallel and dynamic searchable symmetric encryption. *In Financial Cryptography (FC)* , 2013

[9] S. Kamara, C. Papamanthou, and T. Roeder. Dynamic searchable symmetric encryption. In *ACM Conference on Computer and Communications Security* , pages 965–976, 2012

[10] P. van Liesdonk, S. Sedghi, J. Doumen, P. H. Hartel, and W. Jonker. Computationally efficient searchable symmetric encryption. In *Secure Data Management*, pages 87–100, 2010

[11] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In *Advances in Cryptology - EURO-CRYPT'04* , volume 3027 of *Lecture Notes in Computer Science* , pages 506–522. Springer, 2004.

[12] H Hacigumus, B Iyer, S Mehrotra, “Efficient execution of aggregation queries over encrypted relational databases”. DASFAA 2004, LNCS 2793, 125–136 (2004)

[13] E.-J. Goh. 2003. "Secure Indexes," *IACR Cryptology ePrint Archive*, vol. 2003, p. 216

[14] 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*, pp. 79-88.

[15] P.Golle, J.Staddon, B.Waters. Secure conjunctive search over encrypted data. In: ACNS 2004, Lecture notes in computer science, vol.3089. Springer; 2004. pp. 31–45.

[16] Y. Tang, D. Gu, N. Ding, and H. Lu. "Phrase Search over Encrypted Data with Symmetric Encryption Scheme," *in Distributed Computing Systems Workshops (ICDCSW)*, 2012, pp. 471-480.