Volume 1, Issue 9, December 2016 ISSN: 2456-0006
International Journal of Science Technology Management and Research
Available online at: www.ijstmr.com

# Privacy Preserving Data Backup Of Android Device In Multi-Cloud Environment

**Roshan Pundlik Bhusal**
Computer Engineering
Sanghavi College of Engineering
Nashik, India

**Mayur Balasaheb Gatkal**
Computer Engineering
Sanghavi College 0f Engineering
Nashik, India

**Pankaj Gopal Mahajan**
Computer Engineering
Sanghavi College 0f Engineering
Nashik, India

***Abstract:*** ***With the rapid development of cloud computing and mobile networking technologies, users tend to access their stored data from the remote cloud storage with mobile devices. The main advantage of cloud storage is its ubiquitous user accessibility and also its virtually unlimited data storage capabilities. Despite such benefits provided by the cloud, the major challenge that remains is the concern over the confidentiality and privacy of data while adopting the cloud storage services. However, once the data is encrypted, it will limit servers capability for keyword search since the data is encrypted and server simply cannot make a plaintext keyword search on encrypted data. But again we need the keyword search functionality for efficient retrieval of data. To maintain user data confidentiality, the keyword search functionality should be able to perform over encrypted cloud data and additionally it should not leak any information about the searched keyword or the retrieved document. This is known as privacy preserving keyword search. This project aims to study privacy preserving keyword search over encrypted cloud data. Also, we present our implementation of a privacy preserving data storage and retrieval system in cloud computing. For our implementation, we have chosen one of the symmetric key primitives due to its efficiency in mobile environments. The implemented scheme enables a user to store data securely in the cloud by encrypting it before outsourcing and also provides user capability to search over the encrypted data without revealing any information about the data or the query.***

***Keywords: Searchable encryption, Data confidentiality, Cloud storage, Keyword search, Mobile Networking, Shared file systems, Security and reliability issues in Cloud storage applications.***

## I. INTRODUCTION

With the rapid development of cloud computing and mobile networking technologies, users tend to access their stored data from the remote cloud storage with mobile devices. The main advantage of cloud storage is its ubiquitous user accessibility and also its virtually unlimited data storage capabilities. Despite such benefits provided by the cloud, the major challenge that remains is the concern over the confidentiality and privacy of data while adopting the cloud storage services. However, once the data is encrypted, it will limit servers capability for keyword search since the data is encrypted and server simply cannot make a plaintext keyword search on encrypted data. But again we need the keyword search functionality for efficient retrieval of data. To maintain users data confidentiality, the keyword search functionality should be able to perform over encrypted cloud data and additionally it should not leak any information

about the searched keyword or the retrieved document. This is known as privacy preserving keyword search. This project aims to study privacy preserving keyword search over encrypted cloud data. Also, we present our implementation of a privacy preserving data storage and retrieval system in cloud computing. For our implementation, we have chosen one of the symmetric key primitives due to its efficiency in mobile environments. The implemented scheme enables a user to store data securely in the cloud by encrypting it before outsourcing and also provides user capability to search over the encrypted data without revealing any information about the data or the query.

Several researches have addressed the issue of ensuring confidentiality and privacy of cloud data without compromising the user functionality. Here, confidentiality refers to the secrecy of the stored data so that only the client can read the contents of the stored data. To solve the problem of confidentiality, data encryption schemes can come in handy to provide the users with some control over the secrecy of their stored data. This has been adopted by many recent researches which allow users to encrypt their data before outsourcing to the cloud. However, standard encryption schemes will dampen users' searching ability over the stored data, since after encryption a user simply cannot use a plaintext keyword to perform a search anymore and therefore cannot retrieve the contents in an efficient way. The keyword search functionality enables the user to search for a certain keyword on the remote cloud data. Consider a cloud application that consists of a cloud service provider (CSP), and users who store their data on the cloud storage. The users can use a traditional encryption scheme to ensure the confidentiality of the contents. The naïve approach for retrieving encrypted contents related to a certain keyword would require the user to download all the stored data, and then decrypt and perform the search on the local machine. However, this solution is infeasible from a practical point of view, as the user needs to download all the contents rather than the contents containing the searched keyword. For example, consider a scenario where the cloud storage contains 1 GB of user's data, but only 1 MB of data is related to the searched keyword. Using the naïve solution, it is required to retrieve all the 1 GB data, which is inefficient. Alternatively, the user can store a plaintext keyword index in the cloud server and use it while retrieving the data. However, this approach will allow the CSPs to know about the keyword which is not desirable either. Therefore, to ease the data retrieval from a secure cloud, we need a scheme which enables user capability to search over encrypted contents.

## II. PROPOSED SYSTEM

The keyword search functionality enables the user to search for a certain keyword on the remote cloud data. Consider a cloud application that consists of a cloud service provider (CSP), and users who store their data on the cloud storage. The users can use a traditional encryption scheme to ensure the confidentiality of the contents. The naïve approach for retrieving encrypted contents related to a certain keyword would require the user to download all the stored data, and then decrypt and perform the search on the local machine. However, this solution is infeasible from a practical point of view, as the user needs to download all the contents rather than the contents containing the searched keyword. For example, consider a scenario where the cloud storage contains 1 GB of user's data, but only 1 MB of data is related to the searched keyword. Using the naïve solution, it is required to retrieve all the 1 GB data, which is inefficient. Alternatively, the user can store a plaintext keyword index in the cloud server and use it while retrieving the data. However, this approach will allow the CSPs to know about the keyword which is not desirable either. Therefore, to ease the data retrieval from a secure cloud, we need a scheme which enables user capability to search over encrypted contents.

Here we will store encrypted data in multi cloud environment and shamir secret algorithm will be used to convert data in multi parts and shamir secret reconstruction used for recover the original data.

## III. LITRATURE SURVEY

Consider a cloud application that consists of a cloud service provider (CSP), and users who store their data on the cloud storage. The users can use a traditional encryption scheme to ensure the confidentiality of the contents. The Existing system approach for retrieving encrypted contents related to a certain keyword would require the user to download all the stored data, and then decrypt and perform the search on the local machine. For example, consider a scenario where the cloud storage contains 1 GB of user's data, but only 1 MB of data is related to the searched keyword. Using the naïve solution, it is required to retrieve all the 1 GB data. Alternatively, the user can store a plaintext keyword index in the cloud server and use it while retrieving the data. However, this approach will allow the CSPs to know about the keyword which is not desirable either.

The scopes involve the use of searchable encryption algorithms to search for specific keywords within an encrypted content, i.e., without requiring the user to download the database and decrypt its contents before searching can be performed. The proposed solution delegates searching on encrypted data to CSP but with privacy preserved. For the searchable encryption algorithm, we have chosen to implement the adaptively secure Searchable Symmetric encryption (referred as SSE-2 scheme in the original paper) scheme proposed by Curtmola et al. For the rest of the paper, we will use the term SSE-2 scheme or adaptively secure SSE scheme to refer to the proposed method by Curtmola et al. The SSE-2 scheme provides a simple but efficient method to enable searching over encrypted data while preserving the data privacy. The reason behind selecting a private/symmetric key primitive for our implementation mainly lies in the fact that it results in significantly lesser computational overhead when compared to its public/ asymmetric key counterpart and therefore will be more suitable for mobile devices.

This scheme aims to accelerate the search time by looking into the previously searched keywords. For this, the cloud service provider caches the previously searched keywords to avoid the search on all the stored ciphertexts. However, this comes with the tradeoff of large storage overhead. The first searchable encryption scheme based on symmetric key primitive was proposed by Song et al. This provides a non-keyword based solution. However, this scheme can search for only fixed length words and also the search time is linear in document size, since the it needs to scan the whole document to complete the search. The scheme is too slow when searching for a large number of documents. Goh addresses some of the issues of the above scheme by introducing the concept of a secure index. This scheme generates a search index which can be used to locate the encrypted content. Later, the security notions of searchable symmetric encryption were revisited and stronger security definitions were provided by Curtmola et al. This is a very simple scheme based on keyword indexing approach.

## IV. SYSTEM ARCHITECTURE

This research aims to design and develop a privacy preserving data storage and retrieval system in cloud computing. The scopes involve the use of searchable encryption algorithms to search for specific keywords within an encrypted content, i.e., without requiring the user to download the database and decrypt its contents before searching can be performed

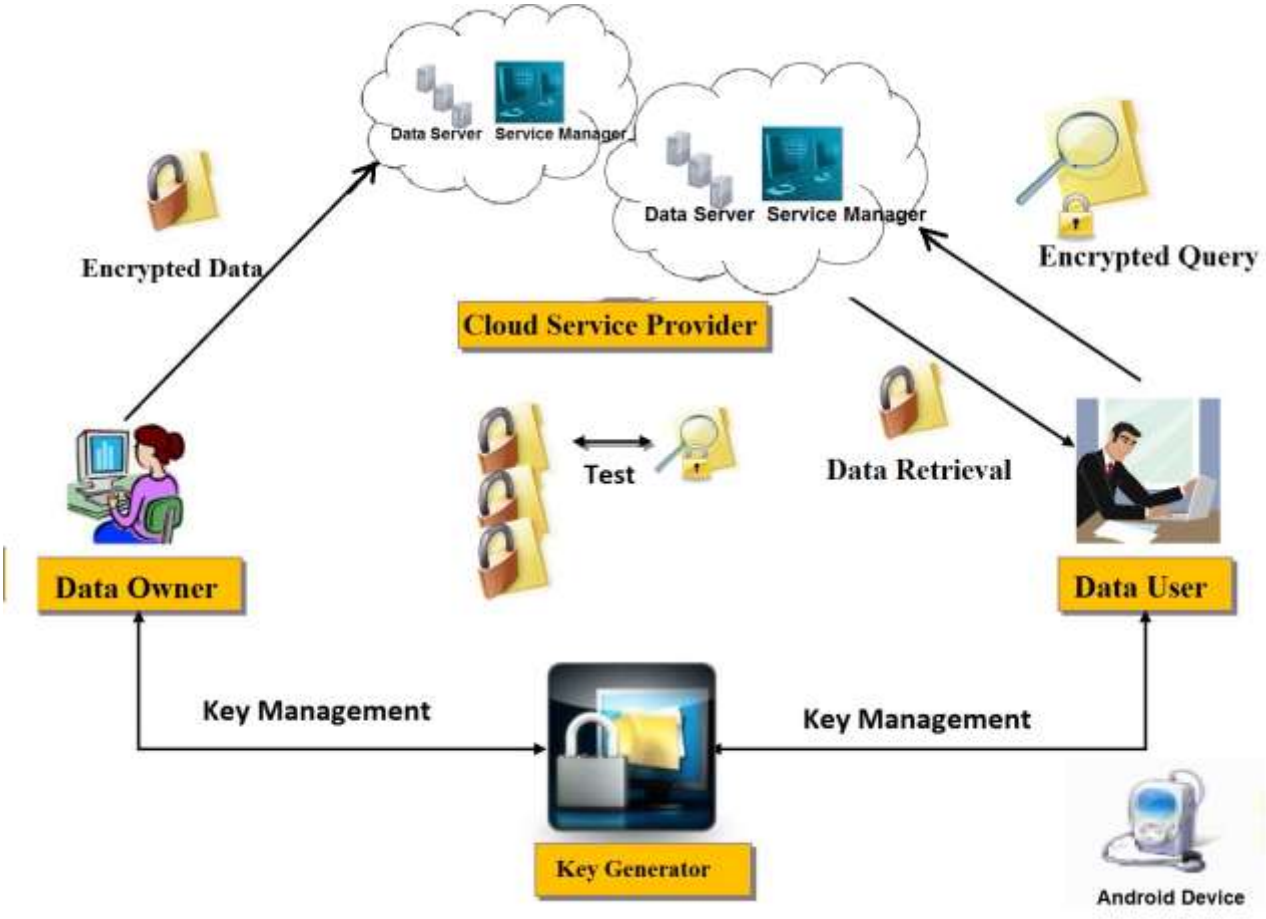


Fig 1: Architecture Diagram

The proposed solution delegates searching on encrypted data to CSP but with privacy preserved. For the searchable encryption algorithm, we have chosen to implement the adaptively secure Searchable Symmetric encryption (referred as SSE-2 scheme in the original paper) scheme proposed by Curtmola et al. For the rest of the paper, we will use the term SSE-2 scheme or adaptively secure SSE scheme to refer to the proposed method by Curtmola et al. The SSE-2 scheme provides a simple but efficient method to enable searching over encrypted data while preserving the data privacy. The reason behind selecting a private/symmetric key primitive for our implementation mainly lies in the fact that it results in significantly lesser computational overhead when compared to its public/ asymmetric key counterpart and therefore will be more suitable for mobile devices.

A. *Key generation*:

It generates the keys for encryption and decryption using a symmetric key primitive. Two keys K1 and K2 are generated for encryption of index contents and documents respectively. Both keys are generated and stored securely at the user device. Note that this is a symmetric key primitive and therefore same key is used for both encryption or decryption purposes.

B. *Document re-processing*:

This block provides the necessary function to pre-process a set of documents and initialize the encryption procedure. Before encrypting the set of documents, the user needs to pre-process the document set to pull out the keyword and build

an index. Let *D* defines a set of document to be encrypted and uploaded in the cloud server. The user has to list out all the keywords from each document in *D* and build an index table listing the documents and the corresponding keywords.

C. *Encryption*:

This block provides the functionality to encrypt the index and document set. The encryption is performed using the encryption keys generated at the key generation step. Basically, it consists of index encryption and document encryptions and performs as described below: *Index encryption* This encrypts the keyword set generated at the pre-processing step and creates an encrypted index/lookup table. The keyword encryption is computed as ENCK1 (wi||ni), where ENCK1 represents encryption with key K1, wi is the keyword *i* and ni is the corresponding document ID containing keyword wi. For each of the encrypted keywords, the encrypted index table lists out the corresponding document ID. For our example document set, the encrypted index table is shown in Table 3. *Document encryption*: This encrypts each document from the document set *D* with key K2, and stores it in the database. The document encryption is computed as ENCK2 (Di) which represents encryption of document Di with key K2. For the above example, the encrypted document lists are shown.

D. *Search token generation*:

This block is used when a user wants to search for a document containing a certain keyword. It provides the user/client with the functionality of generating a search token/ trapdoor which can be used at the server to perform a search over encrypted documents. To carry out the search operation, the user will input the search keyword and then compute the search token.

## CONCLUSION

In this project, we consider how search in encrypted data with different authority by using aggregate key mechanism. This method does not decrypts data at cloud also not allows unauthorised users to access data. Only single aggregate key is required to decrypt and search allocated data. Decryption & search at client side makes System more Secure. Also, the client does not need to maintain a keyword index on its side. However, the index update process is static in our implementation, which does not allow the addition of new files or updating files. On the other hand, our encryption module requires slightly longer time since this also includes the conversion of documents into text file for the keyword extraction process. A faster encryption process can be obtained if the keyword extraction module can work without the document conversion process.

## REFERENCE

1. Kamara S, Lauter K (2010) Cryptographic cloud storage. In: Sion R, Curtmola R, Dietrich S, Kiayias A, Miret JM, Sako K, Sebé F (eds) Financial Cryptography and Data Security, LNCS 6054. Springer, Berlin, Heidelberg, pp 136–149
2. Hacigümüs. H, Iyer B, Li C, Mehrotra S (2002) Executing sql over encrypted data in the database-service-provider model. In: Proceedings of SIGMOD, ACM, pp 216–227
3. Subashini S, Kavitha V (2011) A survey on security issues in service delivery models of cloud computing. J Netw Comput Appl 34:1–11.
4. Kubiatowicz J, Bindel D, Chen Y, Czerwinski S, Eaton P, Geels D et al (2000) Oceanstore: an architecture for globalscale persistent storage. In: Architectural support for programming languages and operating systems, ACM, pp 190–201

5. Muthitacharoen A, Morris R, Gil TM, Chen B (2002) Ivy: a read/write peer-to-peer filesystem. In: Proceedings of the 5th symposium on Operating System Design and Implementation, vol. 36, pp 31–44
6. Adya A, Bolosky WJ, Castro M, Cermak G, Chaiken R, Doucer JR et al (2002) Farsite: federated, available, and reliable storage for an incompletely trusted environment. In: Proceedings of the 5th Symposium on Operating systems design and implementation, vol. 36, pp 1–14
7. Benaloh J, Chase M, Horvitz E, Lauter K (2009) Patient controlled encryption: ensuring privacy of electronic medical records. In: Proceedings of the 2009 ACM workshop on Cloud computing security, ACM, pp 103–114
8. Li M, Lou W, Ren K (2010) Data security and privacy in wireless body area networks. IEEE Wireless Communications Magazine, vol. 17, IEEE, pp 51–58
9. Li M, Yu S, Ren K, Lou W (2010) Securing personal health records in cloud computing: patient-centric and finegrained data access control in multi-owner settings. In: Jajodia S, Zhou J (eds) Security and Privacy in Communication Networks, LNCS 50. Springer, Berlin Heidelberg, pp 89–106
10. Goh EJ (2003) Secure indexes. In: Cryptology ePrint Archive: Report 2003/216
11. Boneh D, Crescenzo GD, Ostrovsky R, Persiano G (2004) Public-key encryption with keyword search. In: Cachin C, Camenisch JL (eds) Advances in Cryptology EUROCRYPT, LNCS 3027. Springer, Berlin Heidelberg, pp 506–522
12. Boneh D, Waters B (2007) Conjunctive, subset, and range queries on encrypted data. In: Salil Vadhan P (ed) Theory of cryptography, LNCS 4392, Springer, Berlin Heidelberg, pp 535–554
13. Liu Q, Wang G, Wu J (2009) An efficient privacy preserving keyword search scheme in cloud computing. In: International Conference on Computational Science and Engineering (CSE), Vol. 2, pp 715–720