International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-10, August 2019

Implementation of User Centric SSE for Privacy-Preserving and Enhancing Searching Efficiency in Cloud Environment

Rudragoud Patil, R. H. Goudar

Abstract**:** ***Cloud computing allows the users geographically located to access data and application from a cloud which follows the pay-as-you-go financial model. This technology provides a method to enhance the capabilities dynamically without having to spend in new infrastructure or training personnel. Cloud computing promises to offer reliable services to the customer through the construction of next-generation data centers and storage technology. With its wide advantages, cloud computing has emerged as fast emergent segments of the IT. Hence data security is a major concern for cloud users. One solution to this is that user can apply encryption techniques on data before sending it to the cloud server. But encrypted data limits the server's capability to make plaintext keyword search. Keyword search functionality that can function over the encrypted data is desirable but no information about the searched keyword or the retrieved document should not compromise in this process. Our implementation of searchable symmetric encryption proves that it is not vulnerable to such things and preserves privacy of user data and query.***

Keywords: Privacy preserving, Keyword search, Encrypted data.

I. INTRODUCTION

Cloud offers storage service to the clients. The data stored on the cloud is actually stored on the remote servers, which can be considered as an extra hard drive. The data stored on cloud can be accessed from anywhere and at any time through a connected device using web pages (Ex: Google docs). The data stored on cloud can be of any type, such as documents, photos, multimedia, etc.

Cloud protects the data being stored from viruses and also damage to any physical devices such as computers. Cloud computing is storing the data as well as accessing the data and programs through the internet instead of the hard drive on our computer.

Cloud services are not just used by business organizations to fulfill their business requirements, but also by individual users to store their personal data.

Cloud can be categorized into three types based on the mode of deployment –

1. *Private Cloud* – It is owned solely by a single organization. It is more secure compared to other types of cloud deployment, but is expensive to be maintained. The owner has the sole access to this storage space and no other person can access the data stored.
2. *Public Cloud* – It is hosted and also maintained by any CSP (cloud service provider) who provides the required services based on pay-and-use policy. It is less secure compared to private cloud. Here, the storage space is shared between multiple cloud users. The data stored can be vulnerable to attacks from unauthorized person.
3. *Hybrid Cloud* – It is blend of both private and public cloud deployments models. The cloud service provider can choose to provide some specific services from both deployments models or make combination of services from both models and provided to its customers.

Cloud provides three types of services –

1. *Software-as-a-Service* – It's service model used primarily for software delivery. Service provider will host apps in this model and these host apps are available via the Internet to clients.
2. *Platform-as-a-Service* – It gives a platform for the end users to build and host their application without having to build and maintain the required infrastructure. Here, different operating systems and associated services are hosted and given as service to end users via the internet with no need installation.
3. *Infrastructure-as-a-Service* – It provides the potential of virtualized machines. The physical resources like networking, storage, hardware are abstracted with can be shared by multiple operating systems and end user environment.

Local, individual device computation is shifted to distributed, virtual, and scalable resources by Cloud Computing. Cloud computing has emerged to be a buzz word in IT industry in the recent past due to its five important characteristics, they are –

1. *On-demand self- service*: The customer has the capability to unilaterally enhance the computing abilities, without the need for humans to interact. The customer benefits from saving his money and time for the maintenance of resources required for the business to flourish.
2. *Broad network access:* It provides standard mechanisms by which multiple, heterogeneous users can use the client platforms over the network. The user can gain access to his data on cloud with any connected device from anywhere in the world.

Manuscript published on 30 August 2019.
***Correspondence Author(s)**

Rudragoud Patil, Research Scholar, VTU RRC, Department of CSE, KLS GIT, Visvesvaraya Technological University, Belagavi, India.
Dr. R. H. Goudar, Department of CSE, Visvesvaraya Technological University, Belagavi, India.

3. *Resource pooling:* It provides various physical and virtual resources resource pooling capabilities, where they are dynamically assigned using the multiple users based on the demand from the consumers. This ensures equal access to all users as per their requirements.
4. *Rapid elasticity:* The different capabilities can be easily and dynamically commensurate with the increase or decrease in demand and they usually appear unlimited to the consumers.
5. *Measured service:* The usage of resources can be easily managed, monitored and controlled by the CSP (cloud service provider).

II. OBJECTIVE

The primary objective of this scheme is to demonstrating privacy preserving of the user's data and encrypted query on cloud using searchable symmetric encryption scheme. This scheme should not gain any information from the keyword searched and the server should not interpret anything about the retrieved document nor should gain knowledge of about the sequence and number of documents searched or accessed by the user. The secondary objective is to enhance the searching efficiency over the encrypted cloud data.

III. RELATED WORK

This section describes some of the related works in the field of searchable symmetric encryption. Boneh et al., in [1], describes the concept of preserving the confidentiality of data. The confidentiality of the user data is maintained by encrypting the data before storing on to the remote cloud servers. But, encrypted data limits the capability of the user to search over it. Hence he proposes the solution for keyword searching over encrypted data. Therefore, it preserves the privacy of the data and also enhances the searching efficiency over the encrypted content. Curtmola et.al, in [5], describes about the SSE scheme that applies encryption techniques on the data before it is outsourced to the cloud server and searching on the encrypted data by proving search queries privately. It introduces the existing notions of security aspects and defines and proposes new security definitions. The earlier work considered the situation where only the owner of the data was able to provide the search query. The paper extends the capability of searching to multiple users or a group of parties, hence, allowing multiple users to access the content.

Boneh D. et. al, in [2], describes the process of public key encryption having keyword search capability wherein a gateway has the authority to search for the keywords over the transmitted without decrypting other parts of the message. The receiver provides the gateway with the intended keyword to be searched in the message. The gateway does not gain any knowledge about other parts of the encrypted data. This technique uses Identity Based Encryption (IBE) for constructing PEKS. But it poses problems since PEKS is harder to construct.

Liu Q et. al, in [4], the author extends the usability and searching capability to multiple servers rather than a single server. It introduces Threshold Public Key Encryption with Keyword Searching (TPEKS) scheme that distributes the searching procedure for encrypted keywords over multiple servers across the network. This technique provides the advantage over preventing keyword guessing that may in turn cause attacks by malicious servers. This scheme is mostly used in distributed cloud environments to enhance security and reliability over the encrypted search query.

Liu Q et. al, in [3] the author discusses about the disadvantages involved in single keyword and multi keyword searching techniques on the encrypted data. In both the cases, documents containing the searched keyword(s) are retrieved, hence increasing the traffic on the network. Also, to match the searched keyword with the contents of the file, the entire must be decrypted first. The paper proposes the solution of ranked based multi-keyword search technique which gives equivalent matched documents in hierarchical order, thereby reducing communication overhead and enhancing searching efficiency.

Chang YC et.al [6] presented different solutions for privacy preserving keyword search scheme under well defined security and privacy requirements, as no public key cryptographic algorithms are used hence it is more efficient.

Security and privacy are more prominent concerns in delivery models of cloud environment [9]. Authors had given ample survey on all security risks faced by cloud computing

Salam, M.I et.al [12] implemented a searchable symmetric encryption scheme using symmetric key algorithm to maintain privacy of user data on cloud. This scheme guarantees user to store the data on cloud after encryption and also makes easy to search operations on encrypted data.

D. V. N. Siva Kumar [13] presented a brief review on all existing searchable encryption with respect to accuracy and security and also provided few remedial solutions for current existing problems like searching for a file in encrypted database which is present on cloud.

Zuojie Deng et.al [14] proposed multiuser searchable encryption where users will encrypt the files before sending to cloud and also allows other users, who are authorized by cloud data owners to perform search operations on these files.

Manju S Nair and Rajasree [15] implemented a scheme where search operations returns only those documents related to query. It also supports both selective sharing and keyword search among the different customers, who are accessing the files without sharing keys and TPA.

Storing data in cloud has become daily business to any user. Before sending the user data to cloud server it is better to encrypt. Authors [16] had proposed sharable ID based encryption with keyword, which allows any users to search for data which belong to data owner by maintaining privacy of data. Searchable Encryption is most widely accepted technique in both industry and academic research. Authors [17] have implemented a scheme called IDcrypt which addresses problems like security and search operations in cloud environment .This scheme adopts key sharing and identity based encryption.

Suleman.J.Nadaf [18] proposed a privacy preserving scheme by encrypting user health information before outsourcing to cloud storage, as well as providing a keyword search system for cloud documents.

IV. IMPLEMENTAION OF PROPOSED SCHEME

A)Algorithm

In cryptography, Encryption is the process that converts a plain text message into some unreadable format that can be decoded exclusively by the authorized users using a secret key. Advanced Encryption Standard (AES) algorithm is widely accepted encryption algorithm for its long key length and strong encryption mechanism.

AES is a block cipher encryption scheme and is based on Substitution-Permutation Network, involves a series of operations such as substitution and shuffling of the bytes. The plain text is divided into 16 bytes (128 bits) blocks represented as a 4X4 matrix. Based on the length of secret key, each block is processed through a number of rounds. AES uses 10, 12 and 14 rounds for 128 bit, 192 bit and 256 bit key lengths respectively.

The AES algorithm includes the following sub-rounds –

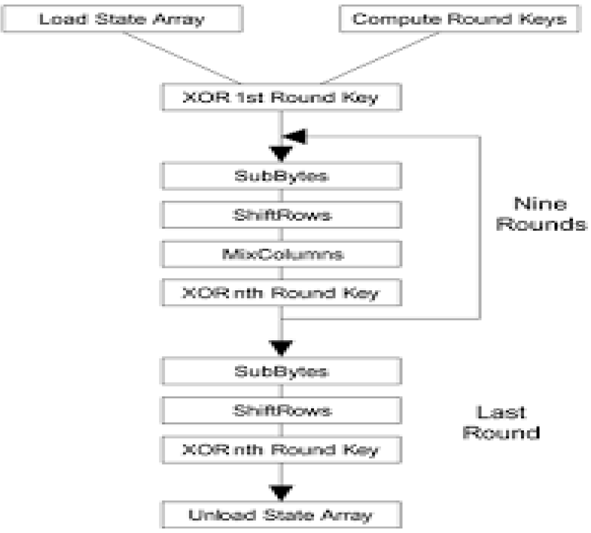

Fig. 1: Structure of AES Algorithm

1. **Byte Substitution:** The 16 bytes of the input are substituted by using the S-box as the lookup table to generate a new set of 16 bytes.
2. **Shift Rows:** All the four rows are shifted in the following way –

First row is unaffected
Second row is shifted to left by one position
Third row is shifted to left by two positions
Fourth row is shifted to left by three positions

3. **Mix Column:** A mathematical function is applied to every column to generate a new matrix. If it is the final round this step is not performed.
4. **Add Round Key:** The 16 bytes are reconsidered as 128 bits which will then be XORed with the 128bits of round key. If this is the last round, then the result is a cipher text, else the process repeats.

B)Working

Our research work details the implementation of our work on using Searchable Symmetric Encryption for preserving privacy of user data and enhancing searching efficiency over the encrypted data. Since most user data kept on the cloud server is confidential, also meant to be kept secret and safe from malicious attacks, the data needs to be converted into some other form before it is outsourced to the cloud servers.

The data is kept confidential by encrypting it before storing it on a remote server. But once the data is encrypted, it becomes difficult to retrieve the data back by using simple search techniques for end users. Our paper proposes a solution for improving the searching efficiency by using keyword search technique.

The implementation uses SSE scheme for encryption and decryption of data. The SSE scheme is going to apply single symmetric key for encryption and decryption, hence it is called Symmetric. The application is demonstrated only for a single user system. Fig. 2 depicts the architecture of our implementation.

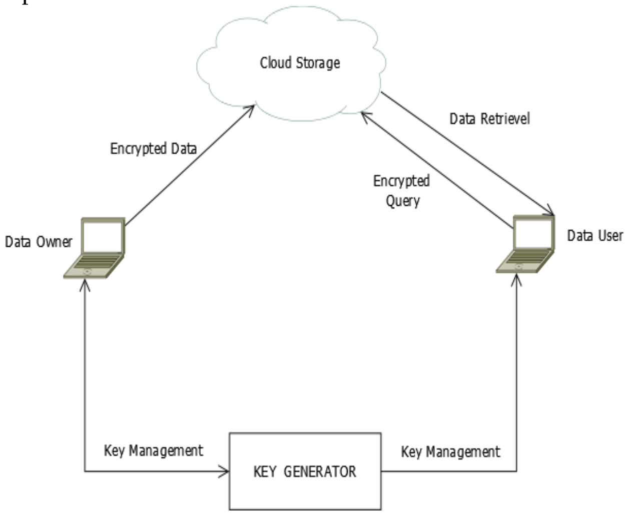

Fig. 2: Proposed System Architecture

It consists of the following components –

1. *Data Owner* – This entity generates the data and encrypts it using a cryptographic scheme before uploading to the cloud storage [7].
2. *Data User* – It is the entity that retrieves the data from the cloud servers by providing search queries. The both data owner and data user are same entities in our implementation.
3. *Cloud Service Provider* – It stores the cipher text documents outsource to the cloud storage by data owner and retrieves the documents/data for the end users. Upon receiving encrypted keyword search query from the user, it does search operations on the corresponding data and retrieves it on behalf of the user from the cloud storage.
4. *Key Generator* – Trusted Third Party that generates and manages the keys that are used do perform encryption and decryption of user data. Our implementation uses only one key per user, to perform cryptographic operations on the data which belongs to end users. The keys are unique for each user.

Searchable Symmetric Encryption

In this section, we describe our proposed Searchable Symmetric Encryption scheme given by Curtmola et al. [5]. In this scheme user U encrypts a set of files M= (F_1, F_2, - - - -, F_n) and makes index file which consist of |M| keywords which are in the form of cipher text. When authorized user/client wants to conduct search operations over encrypted files, U creates trapdoor function and sends it to the cloud server. The responsibility of cloud server is to take this query as input and checks against index file of encrypted keywords which are present in the server. If the appropriate file is located then it retrieves the file containing searched keyword and it is returned to the client. To implement this scheme we have used symmetric key encryption algorithms (AES-128) which is appropriate for a single user scenario. The Searchable Symmetric Encryption scheme mainly consists of five polynomial time functional algorithms: Key generation, Encryption, Trapdoor, Search and Decryption. All of these functions are explained as follows:

1. Keygen: The key generated by using this function used for encryption/decryption operations. Once the key is generated it is stored securely in user machine. In our scheme, we have used symmetric encryption algorithm AES. Keygen() takes security parameter k and returns a secret key *K*.

(K) ← Keygen (k) where K is secret key.

2. Encryption: In this scheme, we encrypt the all the plaintext file documents M= (F_1, F_2, - - - -, F_n) and Keyword associated with each documents with key generated in the previous step.

(C_i) ← Enc_K (F_i)

where in AES encryption, it takes secret key K and Plain text D_i and produces cipher text $C_{i.}$

(Cw_i) ← Enc_K (w_i)

where Enc_K takes Key K and Keyword w_i associated with each file and outputs cipher text of keyword w_i.

3. Trapdoor: This step is required to have search operations. Here user wishes to search for a given file having a specific keyword. Scheme enables any user to generate a encrypted query known as trapdoor and this search is generated by taking encrypted keyword w_i and secret key K.

($T(W_i)$) ← Enc_K (W_i)

4. Search: The search function takes the generated trapdoor token $T(W_i)$ which is cipher text query and checks for this keyword in index of all documents which are present in cloud server.

5. Decrypt: Once user gets requested file, then decrypt the file containing the keyword token. In decryption takes the encrypted file C_i and secret key K and produces the plain text file F_i.

(F_i) ← Enc_K (C_i)

Fig. 3 depicts the detailed data flow diagram. The scheme uses Searchable Symmetric Encryption Scheme proposed by Curtmola et al. [5]. When a new user signs in, a new account is created. A 16-byte (128-bit) secret key is generated for the new user and stored in the database along with other user credentials. This secret key is unique to every user who signs in. This key is necessary to apply cryptographic operations like encryption and decryption on user data whenever the user uploads or downloads the files to or from the cloud respectively.

When the user wishes to upload a new file to the cloud, the application allows the user to browse and select the file from the user's system. The encryption algorithm encrypts both, the contents of the file and the file name, before the file is uploaded to a remote cloud server. The application stores the original file name along with its corresponding encrypted name onto the database as shown in Table 1.

Original File name	Encrypted File name
abc.txt	encrypt7393792920.txt
Test.pdf	encrypt8392759038.pdf
Pic.jpeg	encrypt7604907596.jpeg
Resume.xls	encrypt9302850275.xls
Cloud.doc	encrypt2749017492.doc

Table 1: Index table

The user need not remember the encrypted file name as they are stored and maintained by the application itself. Once the file contents are encrypted, the file is uploaded to the Google Cloud Storage using the gsutil command for file upload. The file remains in the Google Cloud Storage in the encrypted form.

If any authorized user needs to download the file from the cloud storage, the original file name is given as a search query. The application searches for the corresponding encrypted file name from the database. If the file name matches with any of the entries, the encrypted file name is retrieved. This encrypted file name can then be used to download original file from Google Cloud Storage using the gsutil command for file download. The file which is downloaded is decrypted and stored in the user the machine for future use.

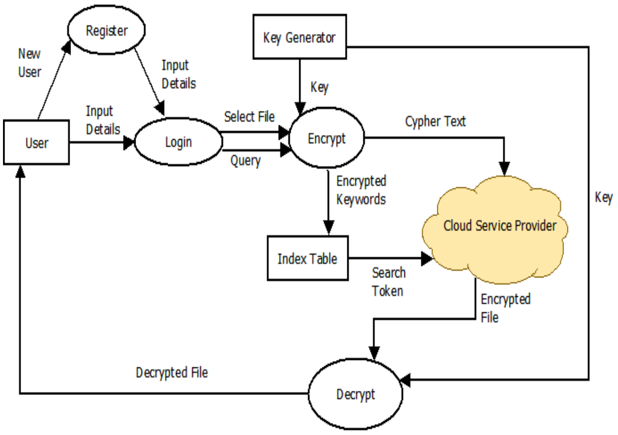

Fig. 3: Data Flow diagram

C)Google Cloud Platform

Google Cloud Platform includes a range of services hosted for compute, storage and application development that run on Google hardware.

It is open to the public domain and accessed by any software professionals and individuals using an Internet connection.

Our application utilizes the services for cloud storage to store the encrypted user data on the Google cloud server. The cloud storage provides following to store and manage data.

1. *Buckets:* In Google Cloud Storage, the basic containers that hold the data are Buckets. A bucket contains everything that is stored in the Cloud Storage. Buckets can only be used to arrange the user data and control access to the data but you cannot nest buckets like directories and folders. The buckets have three important properties, a name that is unique globally, the location where the bucket and its objects are stored and the default storage class for the objects that are stored in the bucket. One should always design storage applications with intensive object operations instead of bucket as there are certain limitations on bucket creation and deletion.

2. *Objects:* Individual pieces of data that are stored in the Cloud Storage are the objects. Any number of objects can be created in a bucket. Every object has two components: object data and object metadata. A file that is stored in Cloud Storage is typically Object data and collection of name-value pairs that describe various object qualities is Object metadata.

It also provides a Google Cloud Shell, which is a command line console, to type and execute the gsutil commands.

Google Cloud Storage provides an array of commands to perform different functions on the objects in the cloud. Some of the commands used are as follows –

Creating Storage Buckets:

The following command creates storage bucket specified by the unique name [BUCKET_NAME] for the specific storage class and specific project.

```
gsutilmb -p [PROJECT_NAME] -c [STORAGE_CLASS] -l [BUCKET_LOCATION] gs://[BUCKET_NAME]/
```

where,

PROJECT_NAME specifies name of the currently working project

STORAGE_CLASS defines the default storage class of the bucket

BUCKET_LOCATION specifies the location where the bucket is created and stored; BUCKET_NAME denotes the name of the bucket being created.

Listing bucket objects:

This command lists the contents, i.e. the names of files, present in the specified bucket.

```
gsutills -r gs://[BUCKET_NAME]/**
```

Uploading objects to the bucket:

The following command uploads the local object to the Google Cloud Storage in the specified bucket.

```
gsutilcp [LOCAL_OBJECT_LOCATION] gs://[DESTINATION_BUCKET_NAME]/
```

where,

LOCAL_OBJECT_LOCATION specifies the location of the object on local system that is to be uploaded.

Downloading objects from the bucket:

The following command downloads the file (or object) from the specified bucket on Google Cloud Storage to the specified location on the local system.

```
gsutilcp gs://[BUCKET_NAME]/[OBJECT_NAME] [OBJECT_DESTINATION]
```

where,

OBJECT_DESTINATION specifies the location on the local system where the downloaded object will be stored.

V. RESUSLT ANALYSIS

The paper has successfully demonstrated preserving privacy of user's data by encrypting it using AES algorithm before outsourcing it to the cloud. This section focuses on timings of different activities happening, such as time required encrypting the file, searching it from index table and decrypting it, in milliseconds with respect to different file sizes in MB.

According to the observations made a graph is plotted to depict this. This graph focuses on timings of different activities such as time required to encrypt, decrypt and search of user's data file in the cloud through the information stored in index table. The timings depend on the file size that is uploaded, downloaded or searched for by the user respectively.

Files (MB)	Encrypt(ms)	Search(ms)	Decrypt(ms)
10	360	3867	248
15	206	2659	252
20	235	4963	344
25	1548	10596	490

Table 2: Result analysis

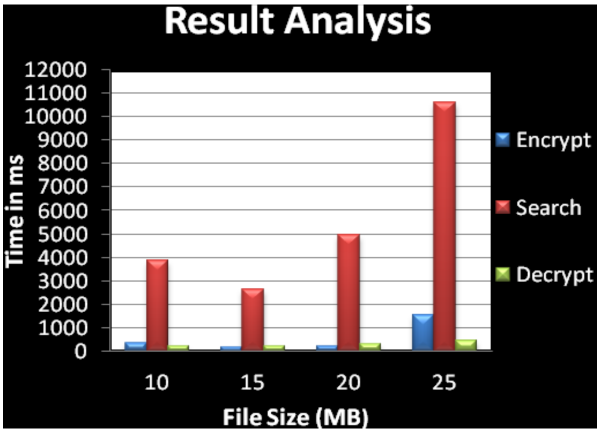

Fig. 4: Graph of Result Analysis

VI. CONCLUSION AND FUTURE SCOPE

The paper provides implementation of Curtmola's SSE scheme. It uses Advanced Encryption Standard (AES) algorithm for encryption and decryption of data files. It provides a detailed workflow of the modules that perform encryption of user's files before sending them to the remote cloud servers.

The AES algorithm provides an effective mechanism to encrypt the data by using a strong secret key. The implementation further improves searching mechanism by enabling the user to perform a search over the encrypted cloud data. Encrypted file name list need not be maintained by the user as the application maintains a database for every file uploaded by the user. The index is updated dynamically whenever the user uploads a new file to the cloud server.

The proposed solution can be extended to support key-sharing mechanism among multiple users whereby one user can gain access to the data file of another user by mutually sharing their secret keys. The implementation can also be enhanced by providing individual users with separate buckets to store their data. This will ensure that two users having a data file with same name can independently store their files on the cloud without facing the conflict of duplicate names.

REFERENCES

1. Boneh D, Waters B (2007) Conjunctive, subset, and range queries on encrypted data. In: SalilVadhan P (ed) Theory ofcryptography, LNCS 4392, Springer, Berlin Heidelberg, pp 535–554.
2. 2.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.
3. Liu Q, Wang G, Wu J (2009) An efficient privacy preserving keyword search scheme in cloud computing. In: InternationalConference on Computational Science and Engineering (CSE), Vol. 2, pp 715–720.
4. Liu Q, Wang G, Wu J (2012) Secure and privacy preserving keyword searching for cloud storage services. J Netw Comput Appl (JNCA) 35(3):927–933.
5. urtmola R, Garay J, Kamara S, Ostrovsky R (2006) Searchable symmetric encryption: improved definitions and efficientconstructions. In: Proceedings of the 13th ACM conference on Computer and communications security, ACM,pp 79–88.
6. Chang YC, Mitzenmacher M (2005) Privacy preserving keyword searches on remote encrypted data. In: Ioannidis J,Keromytis A, Yung M (eds) Applied Cryptography and Network Security, LNCS 3531. Springer, Berlin Heidelberg, pp442–455.
7. 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.
8. 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.
9. Subashini S, Kavitha V (2011) A survey on security issues in service delivery models of cloud computing. J NetwComputAppl 34:1–11.
10. Kubiatowicz J, Bindel D, Chen Y, Czerwinski S, Eaton P, Geels D et al (2000) Oceanstore: an architecture for global-scale persistent storage. In: Architectural support for programming languages and operating systems, ACM, pp.190–201.
11. 11. Muthitacharoen A, Morris R, Gil T M, Chen B (2002) Ivy: a read/write peer-to-peer file system. In: Proceedings of the 5th symposium on Operating System Design and Implementation, vol. 36, pp 31–44.
12. Salam, M.I., Yau, WC., Chin, JJ. et al. Hum. Cent. Comput. Inf. Sci. (2015) 5: 19. https://doi.org/10.1186/s13673-015-0039-9.
13. 13.D. V. N. Siva Kumar and P. Santhi Thilagam “Searchable encryption approaches: attacks and challenges”, Springer-Verlag London Ltd., part of Springer Nature 2018.https://doi.org/10.1007/s10115-018-1309-4.
14. Zuojie Deng-et.al “A multi-user searchable encryption scheme with keyword authorization in a cloud storage” Future Generation Computer Systems, Volume 72, July 2017, Pages 208-218.
15. Manju S Nair and Rajasree M.S “Fine-grained search and access control in multi-user searchable encryption without shared keys” Journal of Information Security and Applications, Volume 41, August 2018, Pages 124-133.
16. Xu, L., Weng, CY., Yuan, LP. et al. J Supercomput (2018) 74: 1001. https://doi.org/10.1007/s11227-015-1515-8
17. 17.G. Wang, C. Liu, Y. Dong, P. Han, H. Pan, B. Fang, "ID-Crypt: A Multi-User Searchable Symmetric Encryption Scheme for Cloud Applications", IEEE Access, vol. 6, pp. 2908-2921, 2018.
18. Suleman. J. Nadaf et.al (2016), “Cloud Based Privacy Preserving Health Data Storage and Retrieval System”, International Conference on Inventive Computation Technologies.

AUTHORS PROFILE

Rudragoud Patil, currently working as an Assistant Professor, Dept of CSE, KLS Gogte Institute of Technology, Belagavi. Research Scholar at VTU-RRC, Belagavi. He has 10 years of Teaching Experience. He has published papers in International Journals and Conferences. His subjects of interest are Cloud Security, Distributed Computing Network Security and Operating Systems.

Dr.R.H. Goudar, currently working as an Associate Professor, Dept. of CNE, Visvesvaraya Technological University, Belagavi. He has 14 years of Teaching Experience at Professional Institutes across India. He worked as a faculty at International Institute of Information Technology, Pune for 4 years and at Indian National Satellite Master Control Facility, Hassan, India. He published over 130 papers in International Journals, Book Chapters and Conferences of High Repute. His Subjects of Interest include Semantic Web, Network Security and Wireless Sensor Networks.