International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 07, July 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

GROUP AND NORMAL DATA SHARING TECHNIQUES THROUGH CLOUD COMPUTING

Dr Y. Ravi Kumar[1], Sivadi Bala Krishna[2], T. Tulasiram[3]

ABSTRACT

Cloud computing is one of the computing techniques which provides communal giving out resources and data to computers and other devices on demand. The ability of selectively distribution encrypted data with special users through public cloud storage may really relieve safety concerns over involuntary data leak in the cloud. Such kind of cases leads to manage efficient data management by designing encryption keys. The single user or group my use different kinds of keys for different documents .The user must submit security keys to share data within the cloud It is a difficult and secured approach to share data via cloud. In this paper we are focusing on this problem, which is more useful to the users, by proposing the concept of key aggregate searchable encryption. In this a data owner only needs to hand out a solo key to a user for sharing documents, and the user only needs to submit a single string to the cloud for querying the shared documents. It is a secured and efficient approach.

Index Terms: **About four key words or phrases in alphabetical order, separated by commas**.

I. INTRODUCTION

The name of cloud computing comes from the ordinary use of a cloud-shaped sign as a concept for the compound communications it contain in system diagram. It entrust isolated military by a user's data, software and computation. Cloud computing consists of hardware and software property complete available on the Internet as managed third-party services. These services naturally give access to higher software applications and high-end networks of attendant comp The aim of cloud computing is to relate usual super computing, or high-performance computing power, normally used by military and research facilities, to perform tens of trillions of computations per second, in consumer-oriented applications such as financial portfolios, to deliver personalized information, to provide data storage or to power large, immersive computer games. The cloud computing uses networks of large groups of servers typically running low-cost consumer PC technology with specialized connections to spread data-processing chores across them. This shared IT infrastructure contains large pools of systems that are linked together. Often, virtualization techniques are used to maximize the power of cloud computing.

Structure of cloud computing

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

II. CHARACTERISTICS AND SERVICES MODELS

The salient characteristics of cloud computing based on the definitions provided by the National Institute of Standards and Terminology (NIST) are outlined below:

- **On-demand self-service**: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service's provider.
- **Broad network access**: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).
- **Resource pooling**: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location-independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or data center). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.

III. EXISTING METHODS

There are some encryption techniques like SSE schemes and PEKS schemes. In contrast to those existing work, in the context of cloud storage, keyword search is a common scenario. In such a scenario, the data owner would like to share a document with group users. Each user who has the access right can provide a trapdoor to perform the keyword search. That procedure is named as the multi-user searchable encryption (MUSE) scenario. Some recent work focuses to such a MUSE scenario, although they all accept single-key joint with access control to achieve the goal. In MUSE schemes, encryption key with all users who can access documents and broadcast encryption is used to achieve common access control. In attribute based encryption (ABE) is used to achieve fine-grained access control aware keyword search. As a result, in MUSE, the main intention is how to control which users can access which documents, whereas how to reduce the number of shared keys and trapdoors is not considered. The disadvantages of existing system include unexpected privilege escalation, it is not efficient and Shared data will not be secure.

IV. PRAPOSED METHOD

In this paper, we are proposing the concept of key-aggregate searchable encryption .By using this user can share data by using group key and own key. It requires efficient key management. Here, a data needs to distribute a single aggregate key to each and every user for sharing any number of files. Next, the user needs to submit a single aggregate trapdoor to the cloud for performing searching of keywords over any number of shared files. In this we are using several algorithms for security parameter setup, key generation, encryption, trapdoor generation, trapdoor adjustment, key extraction and trapdoor testing. We then describe both functional and security .We then instantiate the KASE framework by designing a concrete scheme. After providing detailed constructions for the seven algorithms, we analyze the efficiency of the scheme, and establish its security through detailed analysis. The evaluation confirms our system can meet the performance requirements of practical applications.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

V. ADVANTAGES

- 1. It is more secure.
- 2. Decryption key should be sent via a secure channel and kept secret.
- 3. It is an efficient public-key encryption scheme which supports flexible delegation.
- 4. To the best of our knowledge, the KASE scheme proposed in this paper is the first known scheme that can satisfy requirements.

VI. PROBLEM STATEMENT

Consider a scenario where two employees of a company would like to share some confidential busi-ness data using a public cloud storage service (e.g., dropbox or syncplicity). For instance, Alice wants to upload a large collection of financial documents to the cloud storage, which are meant for the directors of different departments to review. Suppose those documents contain highly sensitive information that should only be accessed by authorised users, and Bob is one of the directors and is thus authorized to view documents related to his department. Due to concerns about potential data leakage in the cloud, Alice encrypts these documents with different keys, and generates keyword ciphertexts based on depart-ment names, before uploading to the cloud storage. Alice then uploads and shares those documents with the directors using the sharing functionality of the cloud storage. In order for Bob to view the documents related to his department, Alice must delegate to Bob the rights both for keyword search over those documents, and for decryption of documents related to Bob's department.

With a traditional approach, Alice must securely send all the searchable encryption keys to Bob. After receiving these keys, Bob must store them securely, and then he must generate all the keyword trapdoors using these keys in order to perform a keyword search. As shown in Fig.1(a), Alice is assumed to have a private document set $fdoc_i g^{n}_{i=1}$, and for each document doc_i, a searchable encryption key k_i is used. Without loss of generality, we suppose Alice wants to share m documents $fdoc_i g^{m}_{i=1}$ with Bob. In this case, Alice must send all the searchable encryption keys $fk_i g^{m}_{i=1}$ to Bob. Then, when Bob wants to retrieve documents containing a keyword w, he must generate keyword trapdoor Tr_i for each document doc_i with key k_i and submit all the trapdoors $fTr_i g^{m}_{i=1}$ to the cloud server. When m is sufficiently large, the key distribution and storage as well as the trapdoor gener-ation may become too expensive for Bob's client-side device, which basically defies the purpose of using cloud storage.

In this paper, we propose the novel approach of key-aggregate searchable encryption (KASE) as a better solution, as depicted in Fig.1(b). , in KASE, Alice only needs to distribute a single aggregate key, instead of $fk_i g^{m}_{i=1}$ for sharing m documents with Bob, and Bob only needs to submit a single aggregate trapdoor, instead of $fTr_i g^{m}_{i=1}$, to the cloud server.

VII. ALGORITHM

In this paper, we propose the novel approach of Key-aggregate searchable encryption (KASE) as a enhanced solution, as depicted in Fig.1(b). , in KASE, seeta needs to issue a single aggregate key, instead of $\{ki\}^{m}i=1$ for sharing m documents with Ram, and ram needs to issue a single aggregate trapdoor, instead of $\{ Tri \}^{m}i=1$, to the cloud server. The cloud server can utilize this aggregate trapdoor and some public data to carry out keyword search and revisit the result to Ram. As a result, in KASE, the delegation of keyword search right can be achieved by

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

sharing the single aggregate key.

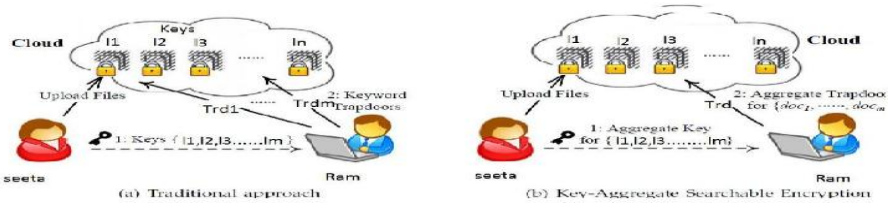


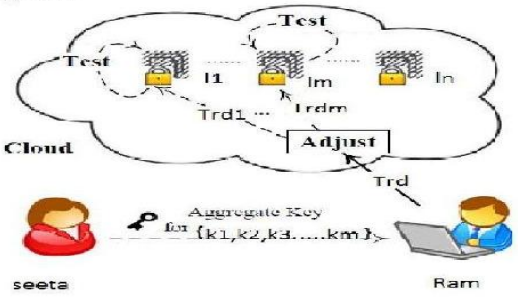
Fig. 1. keyword search in group data sharing system.

To design a key-aggregate searchable encryption method under which any subset of the keyword ciphertexts from any set of documents is searchable with a constant-size trapdoor generated by a constant size aggregate key.

- **The KASE construction**

The KASE construction is composed of several algorithms. Specially, to set up the method, the cloud server would generate public parameters of the system during the **Setup** algorithm, and these public parameters can be reprocess by dissimilar data owners to distribute their files. For each data owner, they should produce a public/master-secret key pair through the **Keygen** algorithm. Keywords of each document can be encrypted through the **Encrypt** algorithm with the exclusive searchable encryption key. In that case, the data owner can apply the master-secret key to produce an aggregate searchable encryption key for a group of selected documents through the **Extract** algorithm. The aggregate key can be spread securely to approve users who

Fig. 2. Framework of key-aggregate searchable encryption.



need to access those documents. After that, as shown in Fig.2, an certified user can create a keyword trapdoor via the **Trapdoor** algorithm using this aggregate key, and submit the trapdoor to the cloud. After getting the trapdoor, to carry out the keyword search over the particular set of documents, the cloud server will run the **Adjust** algorithm to produce the right trapdoor for each document, and then run the **Test** algorithm to test whether the document contains the keyword.

This construction is summarized in the following.

1. **Setup**(1^{λ}, n): This algorithm is run by the cloud service provider to set up the scheme. On input of a security parameter 1^{λ} and the maximum possible number n of documents which belongs to a data owner, it outputs the public system parameter params.
2. **Keygen**: This algorithm is run by the data owner to generate a random key pair (pk,msk).
3. **Encrypt**(pk, i): This algorithm is run by the data owner to encrypt the i-th document and generate its keywords' ciphertexts. For each document, this algorithm will create a delta Дi for its searchable encryption key ki. On input of the owner's public key pk and the file index i, this algorithm outputs data ciphertext and keyword ciphertexts Ci.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

3. **Extract**(msk, S): This algorithm is run by the

data owner to generate an aggregate searchable encryption key for hand over the keyword search right for a certain set of documents to other users.

It takes as input the owner's master-secret key msk and a set S which enclose the directory of documents, and then outputs the aggregate key kagg.

4. **Trapdoor**(kagg, x): This algorithm is run by the user who has the aggregate key to perform a search. It takes as input the aggregate searchable encryption key kagg and a keyword w, then outputs only one trapdoor Trd.

5. **Adjust**(params, i, S, Trd): this algorithm is run by cloud server to adjust the aggregate trapdoor to generate the right trapdoor for each different document. It takes as input the system public parameters params, the set S of documents' indices, the index i of target document and the aggregate trapdoor Tr, then outputs each trapdoor Tri for the i-th target document in S.

6.**Test**(Tri, i): this algorithm is run by the cloud server to perform keyword search over an document.

VIII. IMPLEMENTATION AND RESULT ANALYSIS

It consist of following phases

Data Owner:

In this module we executed by the data owner to setup an account on an untrusted server. On input a security level parameter 1^{λ} and the number of ciphertext classes n (i.e., class index should be an integer bounded by 1 and n), it outputs the public system parameter param, which is omitted from the input of the other algorithms for brevity.

Network Storage (Drop box):

With our solution, Alice can simply send Bob a single aggregate key via a secure e-mail. Bob can download the encrypted photos from Alice's Dropbox space and then use this aggregate key to decrypt these encrypted photos. In this Network Storage is untrusted third party server or dropbox.

Encrypted Aggregate Key and Searchable Encrypted key Transfer:

The data owner establishes the public system parameter via Setup and generates a public/master-secret key pair via KeyGen. Messages can be encrypted via Encrypt by anyone who also decides what ciphertext class is asso-ciated with the plaintext message to be encrypted. The data owner can use the master-secret to generate an aggregate decryption key for a set of ciphertext classes via Extract. The generated keys can be passed to delegates securely (via secure e-mails or secure devices) finally; any user with an aggregate key can decrypt any ciphertext provided that the ciphertext's class is contained in the aggregate key via Decrypt

Trapdoor generation

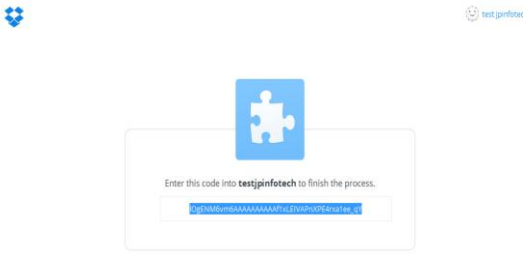
Trapdoor generation algorithm is run by the user who has the aggregate key to perform a search. It takes as input the aggregate searchable encryption key kagg and a keyword w, then outputs only one trapdoor Tr.

File User:

The generated keys can be passed to delegates securely (via secure e-mails or secure devices) finally; any user with the Tappdoor keyword generarion process can decrypt any ciphertext provided that the ciphertext's class is contained in the Encrypted aggregate key and Searchable Encrypted key via Decrypt.

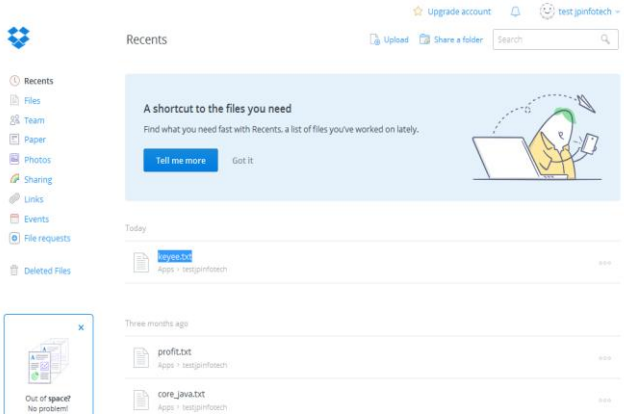
International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 07, July 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

XI. DROP BOX KEY FOR FILE UPLOAD TO CLOUD

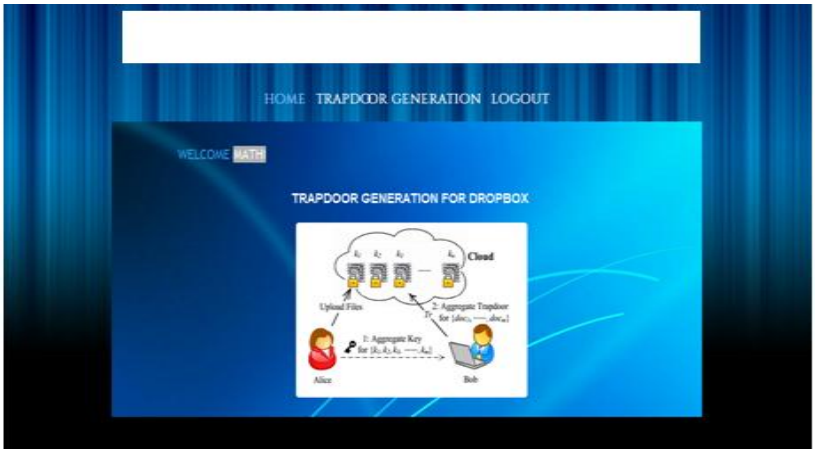


Give a Drop box Key:

File upload to Cloud:

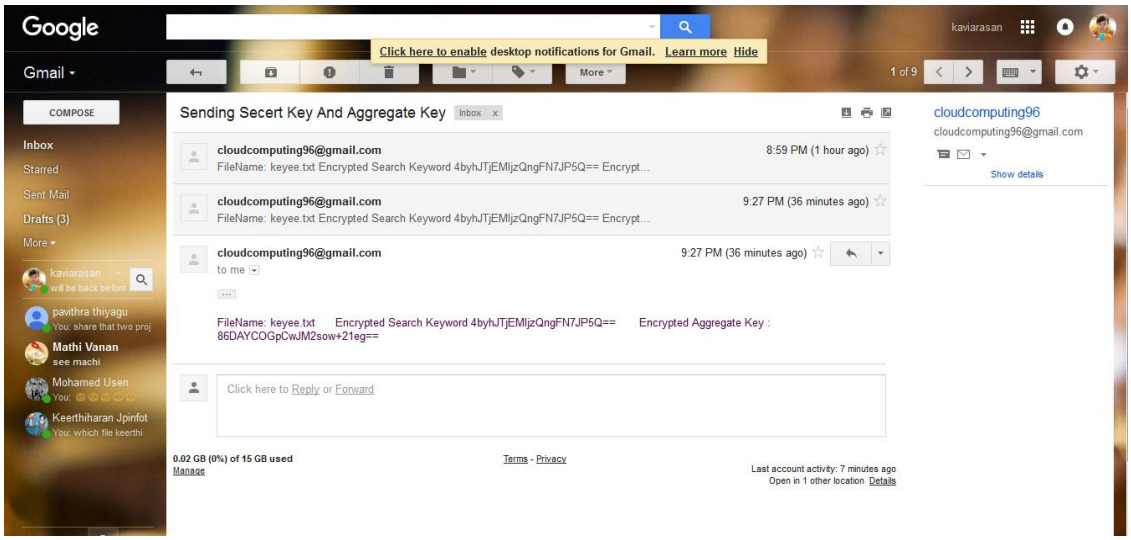


User Home Page:

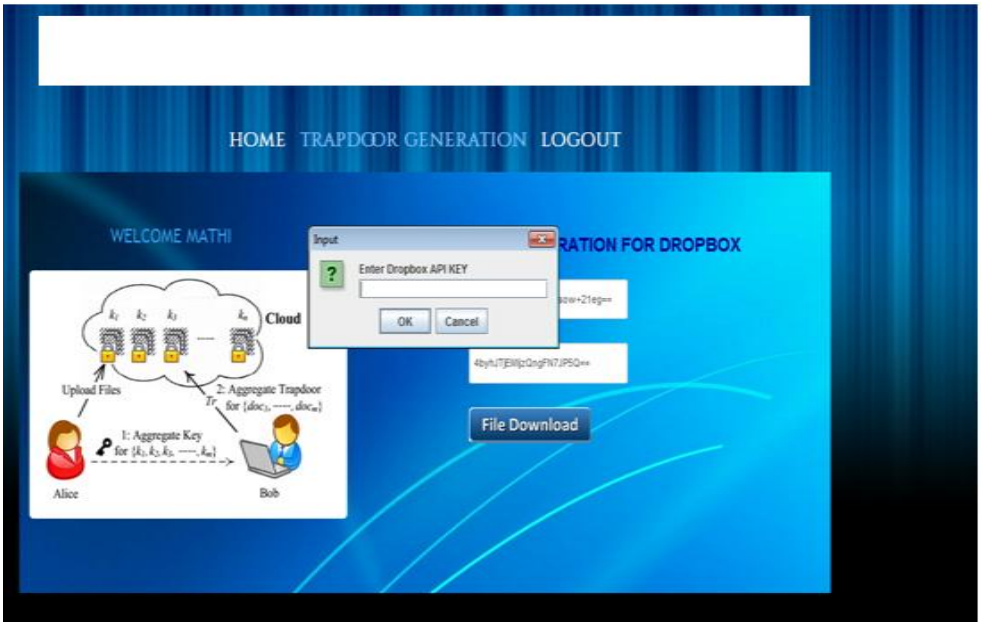


International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 07, July 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

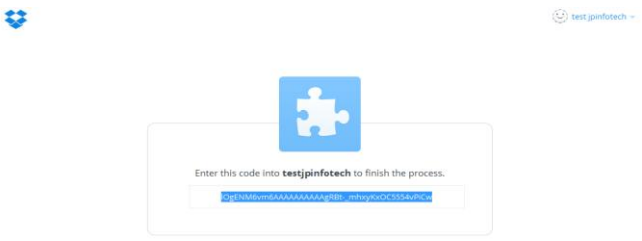
User Get the master key in email:



Give a Drop API Key and next File download:

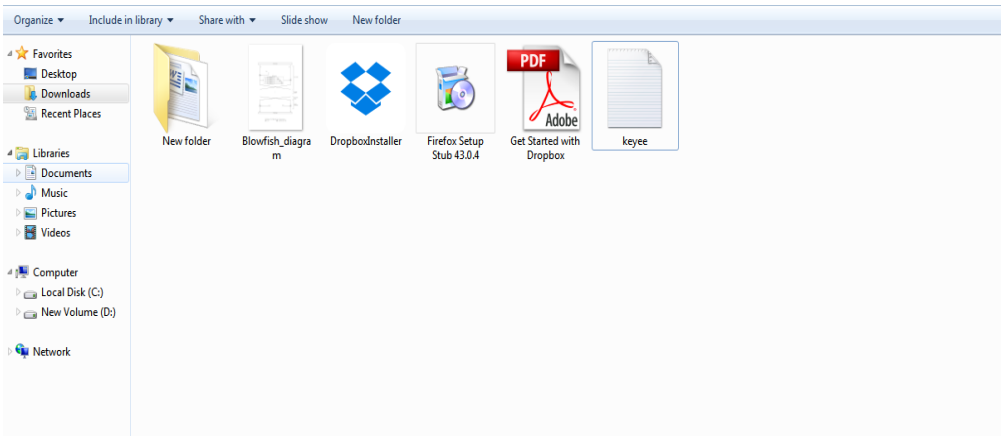


API key:



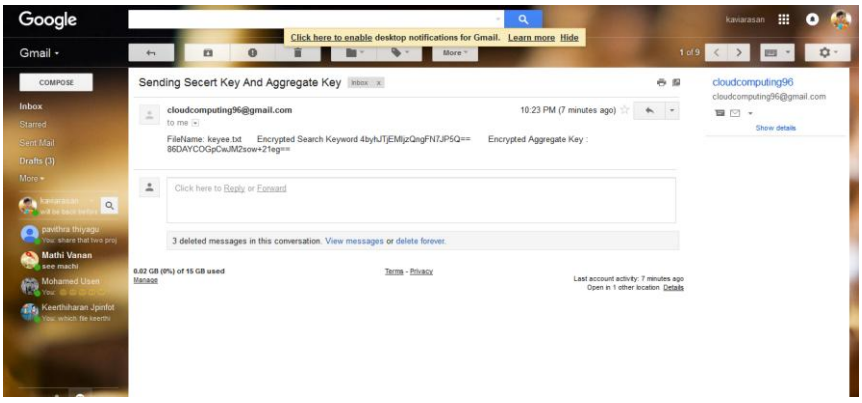
International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 07, July 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

File downloads Folder:



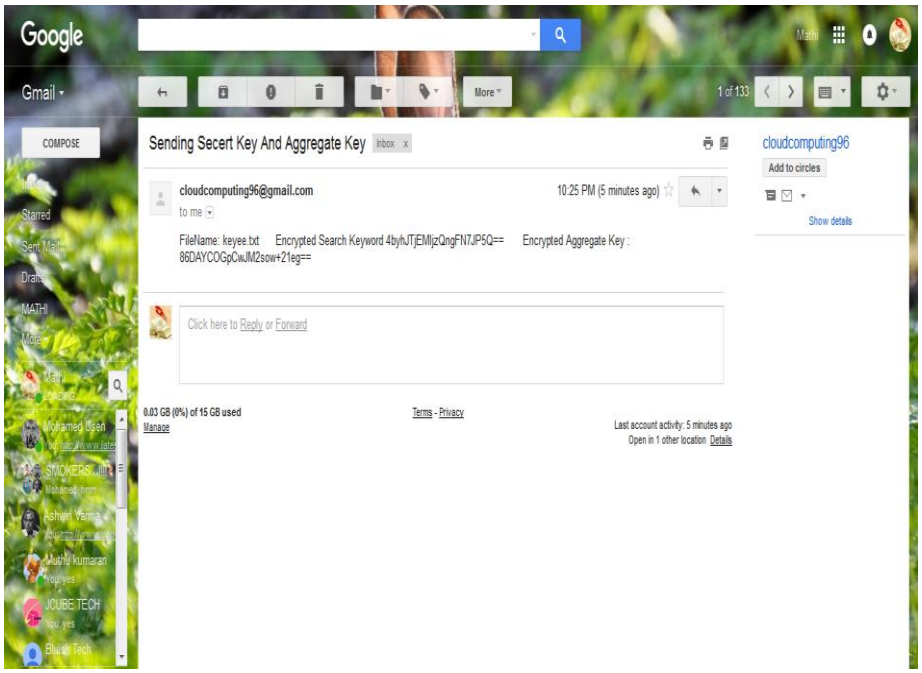
Share the file to Group1 User:

Group1 User:



IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

Group1 User:



X. CONCLUSION

Considering the practical problem of privacy preserving data sharing system based on public cloud storage which requires a data owner to distribute a large number of keys to users to enable them to access his/her documents we for the first time propose the concept of key-aggregate searchable encryption (KASE) and construct a concrete KASE scheme. Both analysis and evaluation results confirm that our work can provide an effective solution to building practical data sharing system based on public cloud storage.

REFERENCES

[1] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing", Proc. IEEE INFOCOM, pp. 534-542, 2010.

[2] R. Lu, X. Lin, X. Liang, and X. Shen, "Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing", Proc. ACM Symp. Information, Computer and Comm. Security, pp. 282-292, 2010.

[3] X. Liu, Y. Zhang, B. Wang, and J. Yan. "Mona: secure multiowner data sharing for dynamic groups in the cloud", IEEE Transactions on Parallel and Distributed Systems, 2013, 24(6): 1182- 1191.

[4] C. Chu, S. Chow,W. Tzeng, et al. "Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage", IEEE Transactions on Parallel and Distributed Systems, 2014, 25(2): 468-477.

[5] X. Song, D.Wagner, A. Perrig. "Practical techniques for searches on encrypted data", IEEE Symposium on Security and Privacy, IEEE Press, pp. 44C55, 2000.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

[6] R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky. "Searchable symmetric encryption: improved definitions and efficient constructions", In: Proceedings of the 13th ACM conference on Computer and Communications Security, ACM Press, pp. 79-88, 2006.

[7] P. Van,S. Sedghi, JM. Doumen. "Computationally efficient searchable symmetric encryption", Secure Data Management, pp. 87-100, 2010.

[8] S. Kamara, C. Papamanthou, T. Roeder. "Dynamic searchable symmetric encryption", Proceedings of the 2012 ACM conference on Computer and communications security (CCS), ACM, pp. 965- 976, 2012.

[9] D. Boneh, C. G, R. Ostrovsky, G. Persiano. "Public Key Encryption with Keyword Search", EUROCRYPT 2004, pp. 506C522, 2004.

1.Dr Y. RAVI KUMAR ,Working as Professor in the department of CSE, Keshav Memorial Institute of Technology,TS,India. He was participated at different national and International Conferences. His research interested in Designing, Networking and Testing and Data Warehousing.

Gmail: ravikumar.yeetha@aecn.in

2.Sivadi Balakrishna Working as Assistant Professor in the department of CSE, Keshav Memorial Institute of Technology,TS, His Research interests on Software Engineering, Software architecture and Web Technologies. He has 2 international publications and 2 international conference Publications at International and National level.

Gmail: balakrishna.sivadi@gmail.com

3. T.TULASIRAM,working as Assistant Professor in the department of CSE, QISCET, ONGOLE. His Research interests on Software Engineering, Software architecture and Web Technologies. He has 2 international conference Publications at International and National level.

Gmail: tulasiramt@gmail.com