Single Key - Explore Encryption (SKEE) for Sharable Data in Cloud Storage

K. Neeharika Reddy[1]; Mahammad Shareef M[2] & Karamala Suresh[3]

[1]PG Student Dept of CSE MJR College of Engineering & Technology Chitoor Road, Near Agraharam, Piler, Andhra Pradesh 517214 neeharika.mca@gmail.com

[2]Assistant Professor Dept of CSE MJR College of Engineering & Technology Chitoor Road, Near Agraharam, Piler, Andhra Pradesh 517214 mekalashareef@gmail.com

[3]Assistant Professor & H.O.D Dept of CSE MJR College of Engineering & Technology Chitoor Road, Near Agraharam, Piler, Andhra Pradesh 517214 ksuresh.madina@gmail.com

Abstract:

The ability of specifically offering scrambled information to diverse clients by means of open distributed storage might extraordinarily ease security worries over unintentional information spills in the cloud. A key test to planning such encryption plans lies in the productive administration of encryption keys. The wanted adaptability of imparting any gathering of chose reports to any gathering of client's requests diverse encryption keys to be utilized for distinctive archives. On the other hand, this additionally infers the need of safely conveying to clients an extensive number of keys for both encryption and seeks, and those clients will need to safely store the got keys, and present a just as substantial number of catchphrase trapdoors to the cloud to perform look over the common information. The inferred requirement for secure correspondence, stockpiling, and multifaceted nature unmistakably renders the methodology unfeasible. In this paper, we address this down to earth issue, which is generally disregarded in the writing, by proposing the novel idea of Single-Key Explore encryption and instantiating the idea through a solid SKEE plan, in which an information proprietor just needs to disseminate a solitary key to a client for sharing countless, and the client just needs to present a solitary trapdoor to the cloud for questioning the mutual reports. The security examination and execution assessment both affirm that our proposed plans are provably secure and for all intents and purposes productive.

Keywords: - SKEE; data privacy & sharing; cloud storage

1. INTRODUCTION

Cloud storage has emerged as a promising solution for providing ubiquitous, convenient, and on-demand accesses to large amounts of data shared over the Internet. Today, millions of users are sharing personal data, such as photos and videos, with their friends through social network applications based on cloud storage on a daily basis. Business users are also being attracted by cloud storage due to its numerous benefits, including lower cost, greater agility, and better resource utilization. However, while enjoying the convenience of sharing data via cloud storage, users are also increasingly concerned about inadvertent data leaks in the cloud. Such data leaks, caused by a malicious adversary or a misbehaving cloud operator, can usually lead to serious breaches of personal privacy or business secrets (e.g., the recent high profile incident of celebrity photos being leaked in I Cloud). To address users' concerns over potential data leaks in cloud storage, a common approach is for the data owner to encrypt all the data before uploading them to the cloud, such that later the encrypted data may be retrieved and decrypted by those who have the decryption keys. Such cloud storage is often called the cryptographic cloud storage. However, the encryption of data makes it challenging for users to search and then selectively retrieve only the data containing given keywords. A common solution is to employ a Explore encryption (SE) scheme in which the data owner is required to encrypt potential keywords and upload them to the cloud together with encrypted data, such that, for retrieving data matching a keyword, the user will send the corresponding keyword trapdoor to the cloud for performing search over the encrypted data.

2. RELATED WORK

Existing System

Consider a scenario where two employees of a company would like to share some confidential business data using a public cloud storage service (e.g., drop box or syncplicity). For instance, User-1 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 authorized users, and User-2 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, User-1 encrypts these documents with different keys, and generates keyword cipher texts based on department names, before uploading to the cloud storage. User-1 then uploads and shares those documents with the directors using the sharing functionality of the cloud storage. In order for User-2 to view the documents related to his department, User-1 must delegate to User-2 the rights both for keyword search over those documents and for decryption of documents related to User-2's department. With a traditional approach, User-1 must securely send all the Explore encryption keys to User-2. After receiving these keys, User-2 must store them securely, and then he must generate all the keyword trapdoors using these keys in order to perform a keyword search. User-1 is assumed to have a private document set {doci}n i=1, and for each document doci, a Explore encryption key kiis used. Without loss of generality, we suppose User-1 wants to share documents {doci}m i=1 with User-2. In this case, User-1 must send all the Explore encryption keys {ki} m i=1 to User-2. Then, when User-2 wants to retrieve documents containing a keyword w, he must generate keyword trapdoor Tri for each document doci with key ki and submit all the trapdoors {Tri}m i=1 to the cloud server. When m is sufficiently large, the key distribution and storage as well as the trapdoor generation may become too expensive for User-2's client-side device, which basically defies the purpose of using cloud storage.

Proposed System:

In this paper, we address this challenge by proposing the novel concept of key-aggregate Explore encryption (SKEE), and instantiating the concept through a concrete SKEE scheme. The proposed SKEE scheme applies to any cloud storage that supports the Explore group data sharing functionality, which means any user may selectively share a group of selected files with a group of selected users, while allowing the latter to perform keyword search over the former.

To support Explore group data sharing the main requirements for efficient key management are twofold. First, a data owner only needs to distribute a single aggregate key (instead of a group of keys) to a user for sharing any number of files. Second, the user only needs to submit a single aggregate trapdoor (instead of a group of trapdoors) to the cloud for performing keyword search over any number of shared files. We first define a general framework of Single Key Explore encryption (SKEE) composed of seven polynomial algorithms for security parameter setup, key generation, encryption, key extraction, trapdoor generation, trapdoor adjustment, and trapdoor testing. We then describe both functional and security requirements for designing a valid SKEE scheme.

We then instantiate the SKEE framework by designing a concrete SKEE scheme. After providing detailed constructions for the seven algorithms, we analyze the efficiency of the scheme, and establish its security through detailed analysis. We discuss various practical issues in building an actual group data sharing system based on the proposed SKEE scheme, and evaluate its performance. The evaluation confirms our system can meet the performance requirements of practical applications.

3. IMPLEMENTATION

Explore Encryption

Data Group Sharing

Explore Encryption:

Setup: This algorithm is run by the owner set up the scheme. It takes as input a security parameter 1 , and outputs the necessary keys.

Encrypt (k; m): This algorithm is run by the owner to encrypt the data and generate its keyword cipher texts. It takes as input the data m, owner necessary keys including Explore encryption key k and data encryption key, outputs data cipher text and keyword cipher texts Cm.

Trpdr(k; w): This algorithm is run by a user generate a trapdoor Tr for a keyword w using key k.

Test (Tr, C): this algorithm is run by the cloud server to perform a keyword search over encrypted data. It takes as input trapdoor Tr and the keyword cipher texts Cm .Outputs whether Cm contains the specified keyword.

Data Group Sharing:

In which a data owner only needs to distribute a single key to a user for sharing a large number of documents, and the user only needs to submit a single trapdoor to the cloud for querying the shared documents.

4. PROJECT ARCHITECTURE

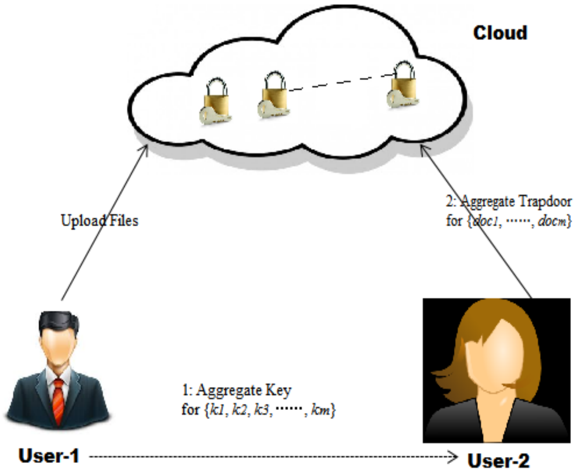
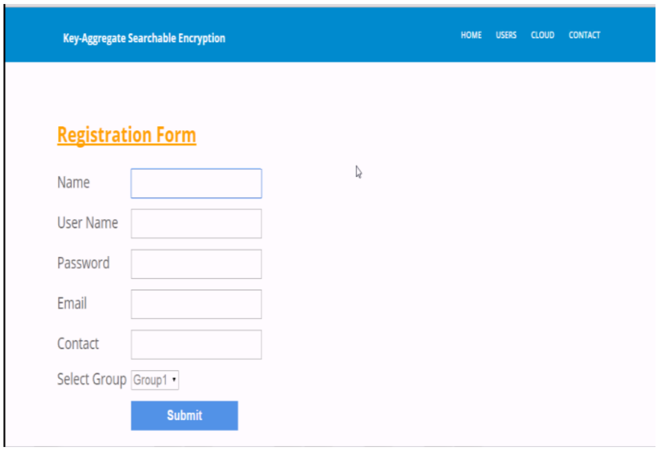


Fig:-1 Project Architecture

The SKEE framework is composed of seven algorithms. Concretely, to establish the scheme, the cloud server would engender public parameters of the system through the Setup algorithm, and these public parameters can be reused by different data owners to apportion their files. For each data owner, he/she should engender a public/master-secret key pair through the Keygen algorithm. Keywords of each document can be encrypted via the Encrypt algorithm with the unique Explore encryption key. Then, the data owner can utilize the master-secret key to engender an aggregate Explore encryption key for a group of culled documents via the Extract algorithm. The aggregate key can be distributed securely (e.g., via secure e-mails or secure contrivances) to sanctioned users who need to access those documents. After that, a sanctioned utilizer can engender a keyword trapdoor via the Trapdoor algorithm utilizing this aggregate key, and submit the trapdoor to the cloud. After receiving the trapdoor, to perform the keyword search over the designated set of documents, the cloud server will run the Adjust algorithm to engender the right trapdoor for each document, and then run the Test algorithm to test whether the document contains the keyword

5. EXPERIMENTAL RESULTS


Key-Aggregate Searchable Encryption

HOME USERS CLOUD CONTACT

Registration Form

Name

User Name

Password

Email

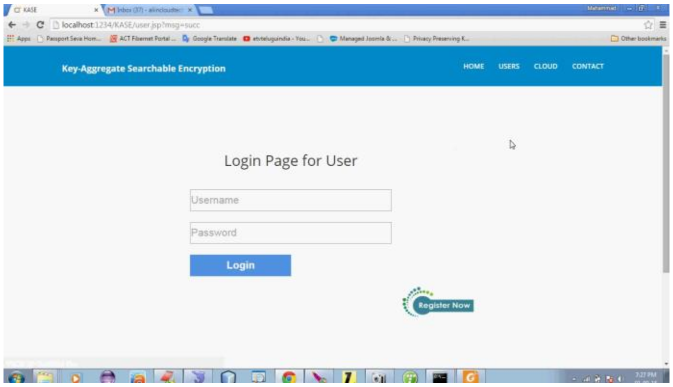
Contact

Select Group Group1

Submit

Fig:-2 Users Registration Page

This screen is illustrating the User Registration page of the application. And also this having fields all user required fields. A New user needs to fill this form for login credentials before entering into application. After successfully registering the user all fields values will be stored into specified database.


localhost:1234/KASE/user.jsp?msg=succ

Key-Aggregate Searchable Encryption

HOME USERS CLOUD CONTACT

Login Page for User

Username

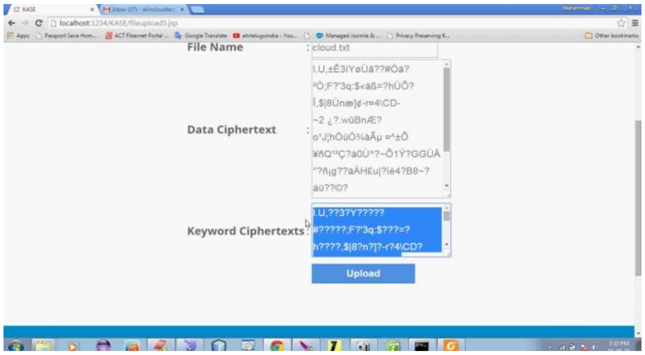
Password

Login

Register Now

Fig:-3 Login page:

This screen is illustrating the User Login page of the application. Here User will give credentials of username and password. If Login credentials' are valid control goes from login page to User Home Page else user login page only display a dialog box of containing Message "Invalid Username OR Password".


localhost:1234/KASE/fileupload5.jsp

File Name : cloud.txt

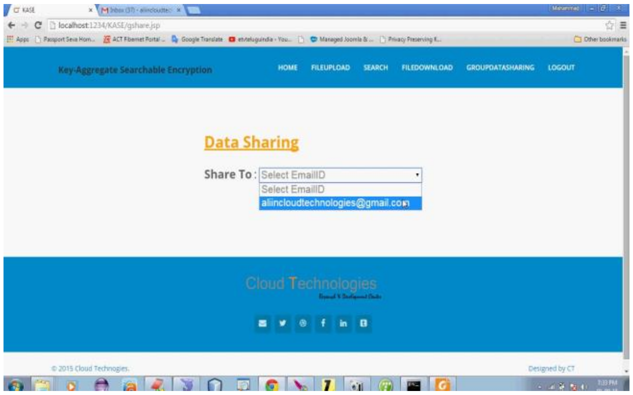
Data Ciphertext : I.U,±Ê3ïYøÙä??#Òä? ²Ó;F?'3q:$«äß=?hÚÕ? Ĩ,$|8Únæ]¢-r¤4\CD- ~2 ¿?.wûBnÆ? o¹J¦hÔûÔ¾äÃµ ¤^±Õ ¥ñQ¹²Ç?á0Ù^?~Õ1Ý?GGÙÅ ¹?ñ¡g??äÀH£u|?ìè4?B8~? äû??©?

Keyword Ciphertexts : I.U,??3?Y????? #?????;F?'3q:$???=? h????,$|8?n?]?-r?4\CD?

Upload

Fig:-4 File with Keyword Ciphertext Page:

File will uploaded along with encrypted format of the system. And also cipher texts of the file will also having filed encrypted format of the application. Except file name file data and keyword cipher texts all having encrypted format for security purpose. Finally having Upload button to upload file.



Data Sharing

Share To : Select EmailID
Select EmailID
aliincloudtechnologies@gmail.com

Fig:-4 Data Sharing Page:

The Above screen describes user sharing the file from user to user of the application

6. 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 Single KeyExplore encryption (SKEE) and construct a concrete SKEE 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. In a SKEE scheme, the owner only needs to distribute a single key to a user when sharing lots of documents with the user, and the user only needs to submit a single trapdoor when he queries over all documents shared by the same owner. However, if a user wants to query over documents shared by multiple owners, he must generate multiple trapdoors to the cloud. How to reduce the number of trapdoors under multi-owners setting is a future work. Moreover, federated clouds have attracted a lot of attention nowadays, but our SKEE cannot be applied in this case directly. It is also a future work to provide the solution for SKEE in the case of federated clouds.

7. REFFERENCESS:

[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 CloudComputing", Proc. ACM Symp. Information, Computer and Comm.
Security, pp. 282-292, 2010.

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

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

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

[6] R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky. "Exploresymmetric 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 efficientExplore

symmetric encryption", Secure Data Management, pp.87-100, 2010.

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

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

[10] Y. Hwang, P. Lee. "Public Key Encryption with ConjunctiveKeyword Search and Its Extension to a Multi-user System", In:Pairing-Based Cryptography C Pairing 2007, LNCS, pp. 2-22, 2007.

Authors Profiles

K. Neeharika Reddy

PG Student
Dept of CSE Mail Id: neeharika.mca@gmail.com
Phone No: - +91 9553172476

Mahammad Shareef M

Assistant Professor
M.Tech,(Ph.D) Dept of CSE Mail Id :- mekalashareef@gmail.com +919642343409

MD SHAREEF currently working as an assistant professor in MJR educational society-pile. He received B. Tech, M. Tech degree from JNTU-A university and he is doing Ph.D in cloud computing area at VIT University-vellore(TN-dist). and his interested research areas are wireless networks, internet of things(IoT), network security.

Karamala Suresh

Assistant Professor& H.O.D
M. Tech Dept of CSE Mail Id: ksuresh.madina@gmail.com Phone No: - +919440503800

Karamala Suresh is a well-disciplined Faculty member. He received B. Tech degree from JNTUA, M. Tech degree from JNTUA. He is not only Management-friendly Faculty but also Student-Friendly Faculty & H.O.D. as a H.O.D he is encouraging the students in to more Research & Development areas by conducting several Technical Fests & Workshops.