Key Aggregate Tagged File Searching (KATFS)

Kiran N.Phulaware[1], Rani B.Gaikwad[2], Prajakta R.Wadekar[3], Pooja R.Balghare[4], Prof. Sonali C Patil[5]
[1, 2, 3, 4, 5] Dept. Of Computer
[1, 2, 3, 4, 5] Siddhant College of Engineering

Abstract- *Data sharing is important functionality in cloud storage .To address user 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. A key challenge to designing such encryption schemes lies in the efficient management of encryption keys. This also implies the necessity of securely distributing to users a large number of keys for both encryption in search and user will have to securely store the received key and submit an equally large number of keywords trapdoors to the cloud in order to perform search over the shared data*

Keywords- roll cage, off-road, dynamic analysis, solid works, hyper mesh

I. INTRODUCTION

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. By addressing this practical problem which is largely neglected in literature, we propose the novel concept of key aggregate Tagged File Searching (KATFS) in which data owner only need to distribute a single key to user for sharing large number of documents and user only needs to submit a single trapdoor to the cloud for querying the shared a large number of documents.

II. RESEARCH WORK

In [1], Kallahalla et al. proposed a cryptographic storage system that enables secure file sharing on untrusted servers, named Plutus. By dividing files into file groups and encrypting each file group with a unique file block key, the data owner can share the file group swith others through delivering the corresponding lock boxkey,where the lock box key issued to encrypt the file block keys.

In [2], files stored on the untrusted server include two parts: file metadata and file data. The file metadata implies the access control information including a series of encrypted key blocks, each of which is encrypted under the public key of authorized users.

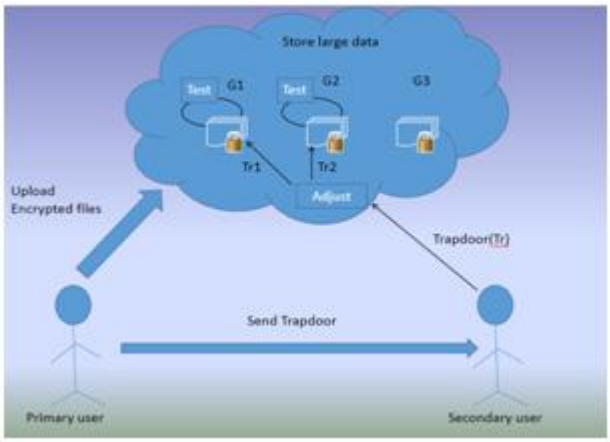
Atenieseetal. [3] average proxy reencryptions to secure distributed storage. specifically, the data owner encrypts blocks of content with unique and symmetric content keys, which are further encrypted under a master public key. For access control, the server uses proxy cryptography to directly reencrypt the appropriate content key(s) from the master public key to a granted users public key.

In [4], Yu et al. presented a scalable and fine grained data access control scheme in cloud computing based on the KPABE technique. The data owner usesar and omkey to encrypt a file, wherether and omkey is further encrypted with a set of attributes using KP-ABE. Then, the group manager assigns an access structure and the corresponding secret key to authorized users, such that a user can only decrypt a cipher text if and only if the data file attributes satisfy the access structure.

III. RESEARCH ELABORATIONS

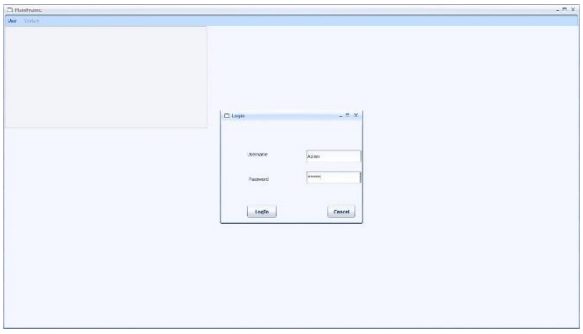
A scenario where two employees of a company would like to share some confidential business data using a public cloud storage service (e.g., drop box). For instance, Primary User wants to upload a large collection of documents to the cloud storage, which are meant for the directors of different departments to review. Suppose those documents contain highly sensitive information that should only be accessed by authorized users, and Secondary User 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, Primary User encrypts these documents with different keys, and generates key word cipher texts based on department names, before uploading to the cloud storage. Primary User then uploads and shares those documents with the directors using the sharing functionality of the cloud storage. In order for Secondary User to view the documents related to his department, Primary User must delegate to Secondary User the rights both for keyword search over those documents, and for decryption of documents related to Secondary Users department. With a traditional approach, Primary Use must securely send all these arch able encryption keys to Secondary User. After receiving these keys, Secondary

User must store them securely, and then he must generate all the key word trap door sousing these key sin order to perform a keyword search. Primary User is assumed to have a private document set and for each document, a searchable encryption key is used. Without loss of generality, we suppose Primary User wants to share m documents with Secondary User. Inthiscase,Primary User must send all these arch able encryptionkeys to Secondary User. Then, when Secondary User wants to retrieve documents containing a keyword, he must generate keyword trapdoor for each document with key and submit all the trap do orsto the cloud server. When missufficiently large, the key distribution and storage as well as the trapdoor generation may become too expensive for Secondary Users client-side device, which basically defines the purpose of using cloud storage.

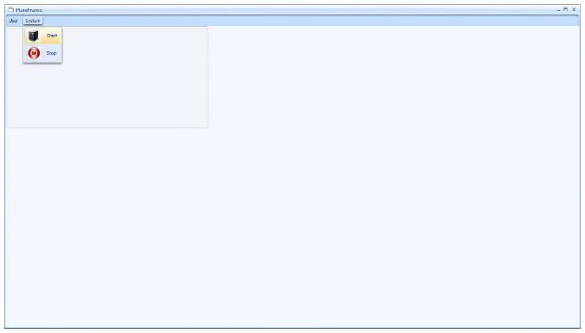


Results:-

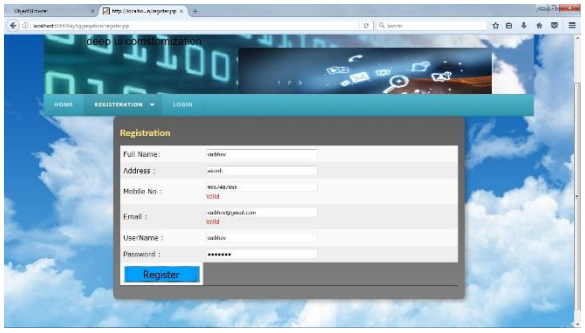
Server Side



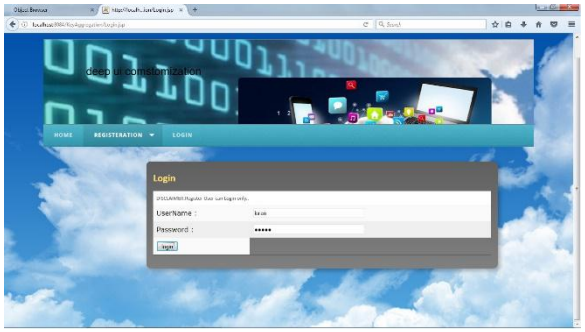
Admin Login



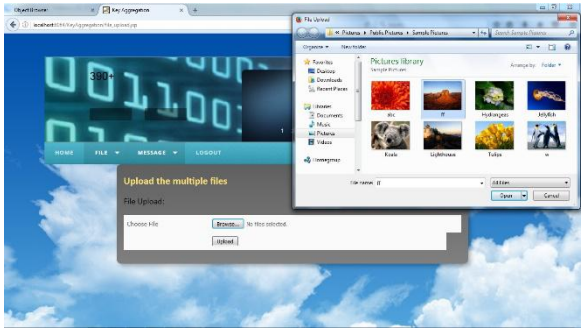
Starting server



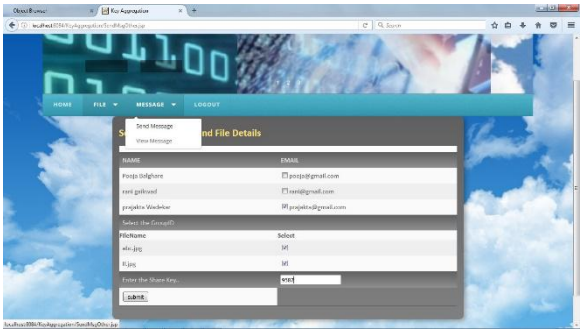
Registration



Login

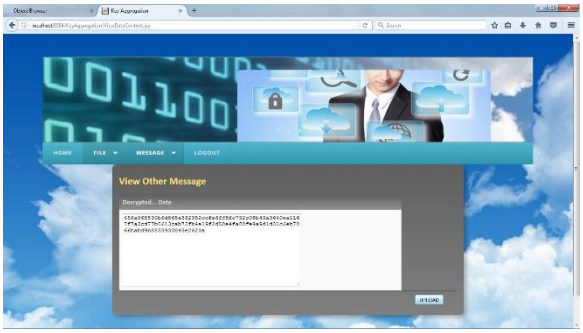


File Uploading

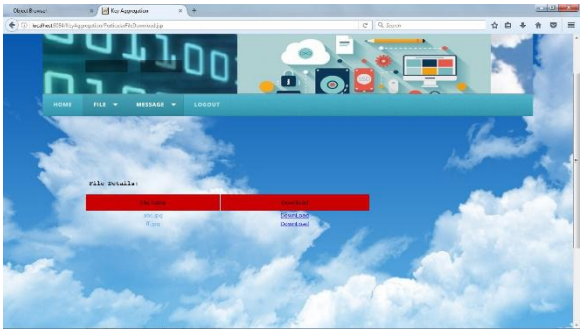


Sending Message

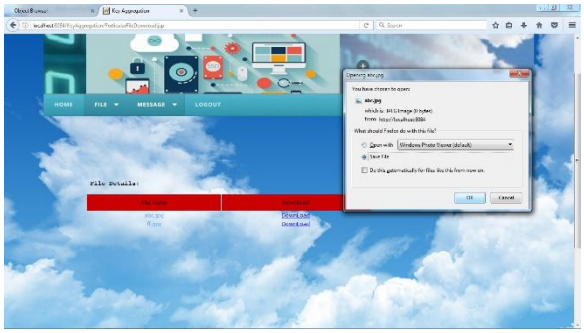
Accessing message



Uploading Trapdoor

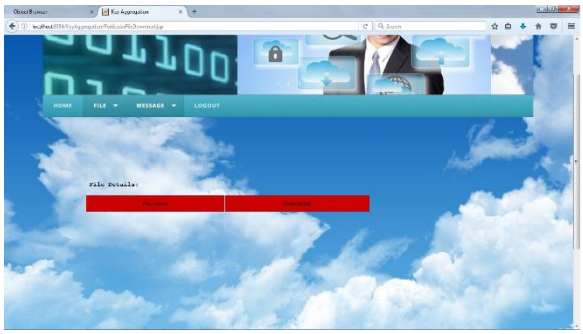


Download File list



Downloading Particular File

If the Trapdoor is copied by another user and want to access the message then Window displayed as



IV. CONCLUSION

In a KATFS 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 overall 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, public clouds have attracted a lot of attention now a days, but our KATFS cannot be applied in this case directly.

V.FUTURESCOPE

How to protect users data privacy is a central question of cloud storage. With more mathematical tools, cryptographic schemes are getting more versatile and often involve multiple keys for a single application. In this article, we consider how to compress secret keys in public-key cryptosystems which support delegation of secret keys for different cipher text classes in cloud storage. Our approach is more flexible than hierarchical key assignment which can only save specific to all key-holders share a similar set of privileges. In cloud storage, the number of cipher texts usually grows rapidly. So we have to reserve enough cipher text classes for the future extension. Otherwise, we need to expand the public-

key. Although the parameter can be downloaded with cipher texts, it would be better if its size is independent of the maximum number of cipher text classes. On the other hand , when one carries the delegated keys around in a mobile device without using special trusted hardware, the key is prompt to leakage, designing a leakage cryptosystem yet allows efficient and flexible key delegation is also an interesting direction.

REFERENCES

[1] Kallahalla,E.Riedel,R.Swaminathan, Q.Wang, and K.Fu, "Plutus: Scalable Secure File Sharing on Untrusted Storage", Proc. USENIX Conf. File and Storage Technologies, pp. 29-42, 2003.

[2] E. Goh, H. Shacham, N. Modadugu, and D. Boneh, "Sirius: Securing Remote Untrusted Storage", Proc. Network and Distributed Systems Security Symp.(NDSS), pp.131-145, 2003.

[3] D. Naor, M. Naor, and J.B. Lotspiech, "Revocation and Tracing Schemes for Stateless Receivers", Proc. Ann. Intl Cryptology Conf. Advances in Cryptology (CRYPTO), pp. 41-62, 2001

[4] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy ReEncryption Schemes with App lications to Secure Distributed Storage", Proc. Network and Distributed Systems SecuritySymp. (NDSS), pp.29-43, 2005

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

[6] 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.