International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/IJSRCSEIT

# Light Weight Secure Data Sharing Scheme with Data Integrity in Cloud Computing

**M. Anjineyulu[1], J. Harathi[1], C. Karthik[1], P. Alekhya[1], V. Chandra Mohan Reddy[1], C. Soundarya[2]**
[1]Student, [2]Assistant Professor
Gates Institute of Technology, Gooty, Andhra Pradesh, India

**Article Info**
Volume 8, Issue 3
Page Number : 412-418
**Publication Issue :**
May-June-2022
**Article History**
Accepted: 10 June 2022
Published: 21 June 2022

## ABSTRACT

With the popularity of cloud computing, mobile devices can store / retrieve personal data anytime, anywhere. As a result, the data security problem in the mobile cloud is exacerbated and prevents further development of the mobile cloud. There are significant studies conducted to improve cloud security. However, most of them do not apply to the mobile cloud as mobile devices only have limited computing resources and power. Mobile cloud applications require a lot of solutions with less computational overhead. In this paper, we propose Lightweight Data Sharing Scheme (LDSS) for mobile cloud computing.

**Keywords:** Security, Integrity, mobile cloud computing, data encryption.

## I. INTRODUCTION

With the development of cloud computing and the popularity of smart mobile devices, people are gradually getting accustomed to a new era of data sharing model in which the data is stored on the cloud and the mobile devices are used to store/retrieve the data from the cloud. Typically, mobile devices only have limited storage space and computing power. On the contrary, the cloud has enormous amount of resources. In such a scenario, to achieve the satisfactory performance, it is essential to use the resources provided by the cloud service provider (CSP) to store and share the data. Nowadays, various cloud mobile applications have been widely used. In these applications, people (data owners) can upload their photos, videos, documents and other files to the cloud and share these data with other people (data users) they like to share. CSPs also provide data management functionality for data owners. Since personal data files are sensitive, data owners are allowed to choose whether to make their data files public or can only be shared with specific data users. Clearly, data privacy of the personal sensitive data is a big concern for many data owners. What is data integrity.

The overall precision, completeness, and continuity of data is known as data integrity. Data integrity also applies to the data's protection and security in terms of regulatory enforcement, such as GDPR compliance. It is kept up to date by a set of procedures, guidelines, and specifications that were put in place during the design phase. It's easy to get the true sense of data integrity muddled because there's so much chatter about it. Data protection and data quality are often

confused with data integrity, but the two terms have different meanings. Data integrity also ensures that the information is protected from outside influences.
Different Kinds of data integrity

- Physical integrity
- Logical integrity
- Entity integrity
- Referential integrity
- Domain integrity
- User-defined integrity

The state-of-the-art privilege management/access control mechanisms provided by the CSP are either not sufficient or not very convenient. They cannot meet all the requirements of data owners. First, when people upload their data files onto the cloud, they are leaving the data in a place where is out of their control, and the CSP may spy on user data for its commercial interests and/or other reasons. Second, people have to send password to each data user if they only want to share the encrypted data with certain users, which is very cumbersome. To simplify the privilege management, the data owner can divide data users into different groups and send password to the groups which they want to share the data. However, this approach requires fine-grained access control. In both cases, password management is a big issue. Apparently, to solve the above problems, personal sensitive data should be encrypted before uploaded onto the cloud so that the data is secure against the CSP. However, the data encryption brings new problems. How to provide efficient access control mechanism on ciphertext decryption so that only the authorized users can access the plaintext data is challenging. In addition, system must offer data owners effective user privilege management capability, so they can grant/revoke data access privileges easily on the data users. There have been substantial researches on the issue of data access control over ciphertext. In these researches, they have the following common assumptions. First, the CSP is considered honest and curious. Second, all the sensitive data are encrypted before uploaded to the Cloud. Third, user authorization on certain data is achieved through encryption/decryption key distribution. In general, we can divide these approaches into four categories: simple ciphertext access control, hierarchical access control, access control based on fully homomorphic encryption and access control based on attribute-based encryption (ABE). All these proposals are designed for non-mobile cloud environment. They consume large amount of storage and computation resources, which are not available for mobile devices. According to the experimental results in [26], the basic ABE operations take much longer time on mobile devices than laptop or desktop computers. It is at least 27 times longer to execute on a smart phone than a personal computer (PC). This means that an encryption operation which takes one minute on a PC will take about half an hour to finish on a mobile device. Furthermore, current solutions don't solve the user privilege change problem very well. Such an operation could result in very high revocation cost. This is not applicable for mobile devices as well. Clearly, there is no proper solution which can effectively solve the secure data sharing problem in mobile cloud. As the mobile cloud becomes more and more popular, providing an efficient secure data sharing mechanism in mobile cloud is in urgent need.

## II. RELATED WORKS

**Implementing Gentry's Fully-Homomorphic Encryption Scheme:** We describe a working implementation of a variant of Gentry's fully homomorphic encryption scheme (STOC 2009), similar to the variant used in an earlier implementation effort by Smart and Vercauteren (PKC 2010). Smart and Vercauteren implemented the underlying "somewhat homomorphic" scheme, but were not able to implement the bootstrapping functionality that is needed to get the complete

scheme to work. We show a number of optimizations that allow us to implement all aspects of the scheme, including the bootstrapping functionality. Our main optimization is a key-generation method for the underlying somewhat homomorphic encryption, that does not require full polynomial inversion. This reduces the asymptotic complexity from O˜(n2.5)O˜(n2.5) to O˜(n1.5)O˜(n1.5) when working with dimension-$n$ lattices (and practically reducing the time from many hours/days to a few seconds/minutes). Other optimizations include a batching technique for encryption, a careful analysis of the degree of the decryption polynomial, and some space/time trade-offs for the fully-homomorphic scheme. We tested our implementation with lattices of several dimensions, corresponding to several security levels. From a "toy" setting in dimension 512, to "small," "medium," and "large" settings in dimensions 2048, 8192, and 32768, respectively. The public-key size ranges in size from 70 Megabytes for the "small" setting to 2.3 Gigabytes for the "large" setting. The time to run one bootstrapping operation (on a 1-CPU 64-bit machine with large memory) ranges from 30 seconds for the "small" setting to 30 minutes for the "large" setting.

**Efficient Fully Homomorphic Encryption from (Standard) LWE:** We present a fully homomorphic encryption scheme that is based solely on the (standard) learning with errors (LWE) assumption. Applying known results on LWE, the security of our scheme is based on the worst-case hardness of "short vector problems" on arbitrary lattices. Our construction improves on previous works in two aspects: 1) We show that "somewhat homomorphic" encryption can be based on LWE, using a new re-linearization technique. In contrast, all previous schemes relied on complexity assumptions related to ideals in various rings. 2) We deviate from the "squashing paradigm" used in all previous works. We introduce a new dimension-modulus reduction technique, which shortens the ciphertexts and reduces the decryption complexity of our scheme, without introducing additional assumptions. Our scheme has very short ciphertexts and we therefore use it to construct an asymptotically efficient LWE-based single-server private information retrieval (PIR) protocol. The communication complexity of our protocol (in the public-key model) is k · polylog(k) + log |DB| bits per single-bit query (here, A; is a security parameter).

**Data leakage mitigation for discretionary access control in collaboration clouds:** With the growing popularity of cloud computing, more and more enterprises are migrating their collaboration platforms from in-enterprise systems to Software as a Service (SaaS) applications. While SaaS collaboration has numerous advantages, it also raises new security challenges. In particular, since SaaS collaboration is increasingly used across enterprise boundaries, organizations are concerned that sensitive information may be leaked to outsiders due to their employees' inadvertent mistakes on information sharing. In this article, we propose to mitigate the data leakage problem in SaaS collaboration systems by reducing human errors. Built on top of the discretionary access control model in existing collaboration systems, we have designed a series of mechanisms to provide defense in depth against information leakage. First, we allow enterprises to encode their organizational security rules as mandatory access control policies, so as to impose coarse-grained restrictions on their employees' discretionary sharing decisions. Second, we design an attribute-based recommender that suggests and prioritizes potential recipients for users' files, reducing errors in the choices of recipients. Third, our system actively examines abnormal recipients entered by a file owner, providing the last line of defense before a file is shared. We have implemented a prototype of our solution and performed experiments on data collected from real-world collaboration systems.

**Secure and efficient access to outsourced data:** Providing secure and efficient access to large scale outsourced data is an important component of cloud computing. In this paper, we propose a mechanism to solve this problem in owner-write-users-read applications. We propose to encrypt every data block with a different key so that flexible cryptography-based access control can be achieved. Through the adoption of key derivation methods, the owner needs to maintain only a few secrets. Analysis shows that the key derivation procedure using hash functions will introduce very limited computation overhead. We propose to use over-encryption and/or lazy revocation to prevent revoked users from getting access to updated data blocks. We design mechanisms to handle both updates to outsourced data and changes in user access rights. We investigate the overhead and safety of the proposed approach, and study mechanisms to improve data access efficiency.

**On key assignment for hierarchical access control:** A key assignment scheme is a cryptographic technique for implementing an information flow policy, sometimes known as hierarchical access control. All the research to date on key assignment schemes has focused on particular encryption techniques rather than an analysis of what features are required of such a scheme. To remedy this we propose a family of generic key assignment schemes and compare their respective advantages. We note that every scheme in the literature is simply an instance of one of our generic schemes. We then conduct an analysis of the Aki-Taylor scheme and propose a number of improvements. We also demonstrate that many of the criticisms that have been made of this scheme in respect of key updates are unfounded, finally, exploiting the deeper understanding we have acquired of key assignment schemes, we introduce a technique for exploiting the respective advantages of different schemes

## III. Methodology

**Proposed system:**

In proposed system, we are implementing data integrity for the cloud data storage. That can be reduces the attacks, and increasing security. The procedure can increases data efficiency.

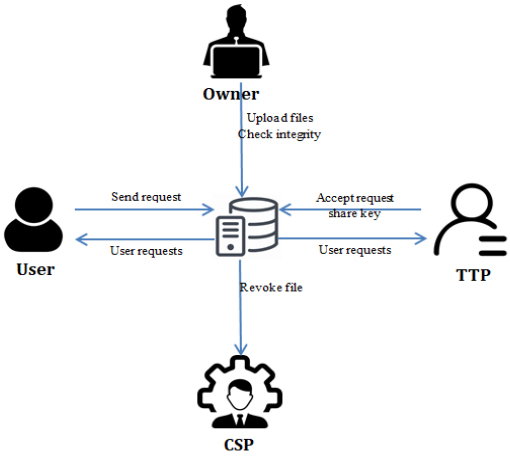



Figure 1 : Block diagram of proposed method

## IV. Implementation

The project has implemented by the process as mentioned below.

**CSP:**

In this module, CSP login into the system with their valid credentials. CSP can able to view owner details and view all file details, and check integrity and he can able to view the user details and then logout from the system.

**TTP:**

In this module, TTP login into the system with their valid credentials. TTP can view requested files, and view users and then logout from the system.

**Data Owner:**

Data Owner can register and login into the system with their credentials. After login he can able to upload the files into cloud, view files and check integrity then logout from the system.

**Data User:**

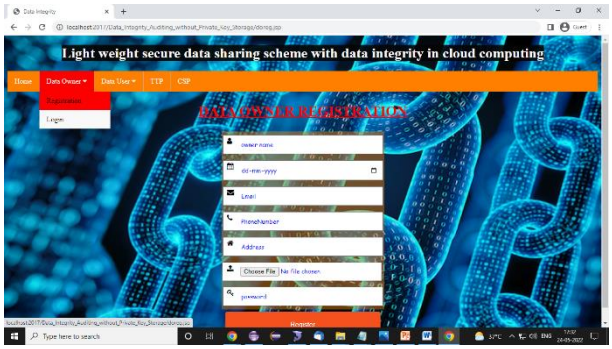
In this module, users are register into the system and then he can able to login with email authentication OTP into the system. After login he can able to search

for files and then user can able download the files and then logout from the system.

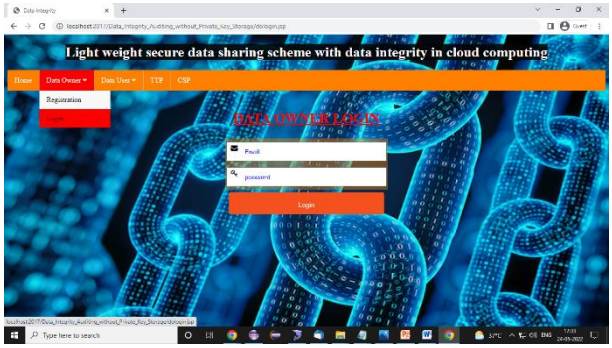
## V. Results and Discussion

**Home page:** This is the home page of the project and it gives us a brief introduction of project.


Data owner registeration

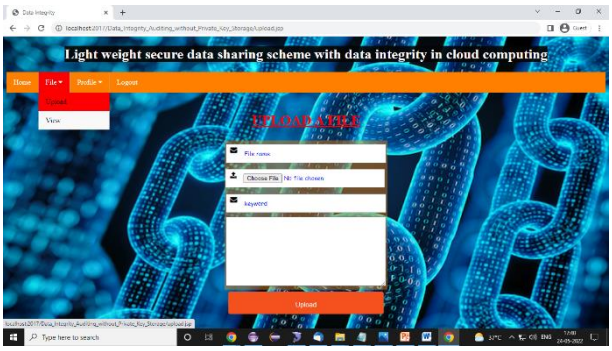



Data owner ogin




Owner home


Upload file




View files


View file data


new data users

data users


files


revoke file


## VI. CONCLUSION

In this project, In proposed system, we are implementing data integrity for the cloud data storage. That can be reduces the attacks, and increasing security. The procedure can increases data efficiency. The integrity check after file sharing, user can get a secret key. Then user can download file by using the secret key, if user submit the wrong key more than 2 time , like the received user was identify as not honest person, then the shared file will be have few changes. Then file owner check the integrity, then it was not secure then apply to revoke the file. Then gets their file status from authority person. If received user can retrieve the file honestly then the file can be secure.

## VII. REFERENCES

[1]. Gentry C, Halevi S. Implementing gentry's fully-homomorphic encryption scheme. in: Advances in Cryptology–EUROCRYPT 2011. Berlin, Heidelberg: Springer press, pp. 129-148, 2011.

[2]. Brakerski Z, Vaikuntanathan V. Efficient fully homomorphic encryption from (standard) LWE. in: Proceeding of IEEE Symposium on Foundations of Computer Science. California, USA: IEEE press, pp. 97-106, Oct. 2011.

[3]. Qihua Wang, Hongxia Jin. "Data leakage mitigation for discertionary access control in collaboration clouds". the 16th ACM Symposium on Access Control Models and Technologies (SACMAT), pp.103-122, Jun. 2011.

[4]. Adam Skillen and Mohammad Mannan. On Implementing Deniable Storage Encryption for Mobile Devices. the 20th Annual Network and Distributed System Security Symposium (NDSS), Feb. 2013.

[5]. Wang W, Li Z, Owens R, et al. Secure and efficient access to outsourced data. in: Proceedings of the 2009 ACM workshop on Cloud computing security. Chicago, USA: ACM pp. 55-66, 2009.

[6]. Maheshwari U, Vingralek R, Shapiro W. How to build a trusted database system on untrusted storage. in: Proceedings of the 4th conference on Symposium on Operating System Design & Implementation-Volume 4. USENIX Association, pp. 10-12, 2000.

[7]. Kan Yang, Xiaohua Jia, Kui Ren: Attribute-based fine-grained access control with efficient revocation in cloud storage systems. ASIACCS 2013, pp. 523-528, 2013.

[8]. Crampton J, Martin K, Wild P. On key assignment for hierarchical access control. in: Computer Security Foundations Workshop. IEEE press, pp. 14-111, 2006.

[9]. Shi E, Bethencourt J, Chan T H H, et al. Multi-dimensional range query over encrypted data. in: Proceedings of Symposium on Security and Privacy (SP), IEEE press, 2007. 350- 364

[10]. Cong Wang, Kui Ren, Shucheng Yu, and Karthik Mahendra Raje Urs. Achieving Usable and Privacy-assured Similarity Search over Outsourced Cloud Data. IEEE INFOCOM 2012, Orlando, Florida, March 25-30, 2012

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

[12]. Kan Yang, Xiaohua Jia, Kui Ren, Bo Zhang, Ruitao Xie: DACMACS: Effective Data Access Control for Multiauthority Cloud Storage Systems. IEEE Transactions on Information Forensics and Security, Vol. 8, No. 11, pp.1790-1801, 2013.

[13]. Stehlé D, Steinfeld R. Faster fully homomorphic encryption. in: Proceedings of 16th International Conference on the Theory and Application of Cryptology and Information Security. Singapore: Springer press, pp.377-394, 2010.

[14]. [14] Junzuo Lai, Robert H. Deng ,Yingjiu Li ,et al. Fully secure keypolicy attribute-based encryption with constant-size ciphertexts and fast decryption. In: Proceedings of the 9th ACM symposium on Information, Computer and Communications Security (ASIACCS), pp. 239-248, Jun. 2014.

**Cite this article as :**

M. Anjineyulu, J. Harathi, C. Karthik, P. Alekhya, V. Chandra Mohan Reddy, C. Soundarya, "Light Weight Secure Data Sharing Scheme with Data Integrity in Cloud Computing", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 8 Issue 3, pp. 412-418, May-June 2022.
Journal URL : https://ijsrcseit.com/CSEIT2283110