ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

IJESRT
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

# BOOSTING EFFICIENCY AND SECURITY FOR CLOUD STORAGE TO OUTSOURCED DATA USING SECURE SHARING SCHEME

**X.Nancy*, S.Selvanayaki**
M.E Student, Assistant Professor, Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Chennai, Tamil Nadu, India

## ABSTRACT
Deduplication is a process in which the duplicate copies of data are eliminated. It is being used in cloud storage for reducing the storage space. The convergent encryption is used for handling the deduplication that manage the their keys in a efficient and secured manner. In this paper, the key management is achieved efficiently. Initially, in baseline approach the master keys are to be protected by each user since they hold independent master keys. Inorder to avoid this, Dekey is implemented in which user only distribute the convergent key instead of managing the master key. As a proof of concept, Secure sharing scheme is used in implementing Dekey in which secret is divided and shared among users.

**KEYWORDS**: convergent encryption, key management, Secure sharing scheme

## INTRODUCTION
The advent of cloud storage motivates enterprises and organizations to outsource data storage to third-party cloud providers, as evidenced by many real-life case studies [3]. One critical challenge of today's cloud storage services is the management of the ever-increasing volume of data [2]. According to the analysis report of IDC, the volume of data in the wild is expected to reach 40 trillion gigabytes in 2020 [9]. To make data management scalable, deduplication has been a well-known technique to reduce storage space [4] and upload bandwidth in cloud storage. Instead of keeping multiple data copies with the same content, deduplication eliminates redundant data by keeping only one physical copy and referring other redundant data to that copy. Each such copy can be defined based on different granularities: it may refer to either a whole file (i.e., filelevel deduplication), or a more fine-grained fixed-size or variable-size data block (i.e., block-level deduplication). Today's commercial cloud storage services, such as Dropbox, Mozy, and Memopal, have been applying deduplication to user data to save maintenance cost [12].

From a user's perspective, data outsourcing raises security and privacy concerns. We must trust third-party cloud providers to properly enforce confidentiality, integrity checking, and access control mechanisms against any insider and outsider attacks. However, deduplication, while improving storage and bandwidth efficiency, is incompatible with traditional encryption. Specifically, traditional encryption requires different users to encrypt their data with their own keys. Thus, identical data copies of different users will lead to different ciphertexts, making deduplication impossible.

Convergent encryption [8] provides a viable option to enforce data confidentiality while realizing deduplication. It encrypts/decrypts a data copy with a convergent key, which is derived by computing the cryptographic hash value of the content of the data copy itself [8]. After key generation and data encryption, users retain the keys and send the ciphertext to the cloud. Since encryption is deterministic, identical data copies will generate the same convergent key and the same ciphertext. This allows the cloud to perform deduplication on the ciphertexts. The ciphertexts can only be decrypted by the corresponding data owners with their convergent keys.

To understand how convergent encryption can be realized, we consider a baseline approach that implements convergent encryption based on a layered approach. That is, the original data copy is first encrypted with a convergent key derived by the data copy itself, and the convergent key is then encrypted by a master key that will be kept locally and securely by each user. The encrypted convergent keys are then stored, along with the corresponding encrypted data copies, in cloud storage. The master key can be used to recover the encrypted keys and hence the encrypted files. In this way, each user only needs to keep the master key and the metadata about the outsourced data [5].

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

However, the baseline approach suffers two critical deployment issues. First, it is inefficient, it will generate an enormous number of keys with the increasing number of users. Specifically, each user must associate an encrypted convergent key with each block of its outsourced encrypted data copies, so as to later restore the data copies. Although different users may share the same data copies, they must have their own set of convergent keys so that no other users can access their files. As a result, the number of convergent keys being introduced linearly scales with the number of blocks being stored and the number of users. This key management overhead becomes more prominent if we exploit fine-grained block-level deduplication. For example, suppose that a user stores 1 TB of data with all unique blocks of size 4 KB each, and that each convergent key is the hash value of SHA-256, which is used by Dropbox for deduplication [15]. Then the total size of keys will be 8 GB. The number of keys is further multiplied by the number of users. The resulting intensive key management overhead leads to the huge storage cost, as users must be billed for storing the large number of keys in the cloud under the pay-as-you-go model.

Second, the baseline approach is unreliable, as it requires each user to dedicatedly protect his own master key. If the master key is accidentally lost, then the user data cannot be recovered; if it is compromised by attackers, then the user data will be leaked.

This motivates us to explore how to efficiently and reliably manage enormous convergent keys, while still achieving secure deduplication. To this end, we propose a new construction called Dekey, which provides efficiency and reliability guarantees for convergent key management on both user and cloud storage sides. Our idea is to apply deduplication to the convergent keys and leverage secret sharing techniques. Specifically, we construct secret shares for the convergent keys and distribute them across multiple independent key servers. Only the first user who uploads the data is required to compute and distribute such secret shares, while all following users who own the same data copy need not compute and store these shares again. To recover data copies, a user must access a minimum number of key servers through authentication and obtain the secret shares to reconstruct the convergent keys. In other words, the secret shares of a convergent key will only be accessible by the authorized users who own the corresponding data copy. This significantly reduces the storage overhead of the convergent keys and makes the key management reliable against failures and attacks. To our knowledge, none of existing studies formally address the problem of convergent key management. This paper makes the following contributions.

. A new construction Dekey is proposed to provide efficient and reliable convergent key management through convergent key deduplication and secret sharing. Dekey supports both file-level and blocklevel deduplications.

. Security analysis demonstrates that Dekey is secure in terms of the definitions specified in the propose security model. In particular, Dekey remains secure even the adversary controls a limited number of key servers.

. We implement Dekey using the Secure sharing scheme that enables the key management to adapt to different reliability and confidentiality levels. Our evaluation demonstrates that Dekey incurs limited overhead in normal upload/download operations in realistic cloud environments.

Secure sharing scheme is a form of secret sharing, where a secret is divided into parts, giving each participant its own unique part, where some of the parts or all of them are needed in order to reconstruct the secret [6]. Counting on all participants to combine together the secret might be impractical, and therefore sometimes the *threshold scheme* is used where any $k$ of the parts are sufficient to reconstruct the original secret.

**OBJECTIVE**

To make data management scalable, deduplication has been a well-known technique to reduce storage space and upload bandwidth in cloud storage. The ciphertexts created can only be decrypted by the corresponding data owners with their convergent keys.

**SCOPE**

To efficiently and securely manage deduplication in cloud using convergent key

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

## SYSTEM ANALYSIS

System Analysis is a process in which the system is analyzed based on the problem domain characteristics.

### EXISTING SYSTEM

In the existing system, using traditional encryption, different users will simply encrypt identical data copies with their own keys, but this will lead to different ciphertexts [14] and hence make deduplication impossible. There are also several implementations of convergent implementations of different convergent encryption variants [1] for secure deduplication It is known that some commercial cloud storage providers, convergent encryption leads to a significant number of convergent keys, but they do not address how to minimize the key management overhead.

**Drawbacks:**

- The problem with secret keys is exchanging them over the Internet or a large network while preventing them from falling into the wrong hands because anyone who knows the secret key can decrypt the message.
- This system do not consider data privacy.

### PROPOSED SYSTEM

We propose a Convergent encryption [8] provides a viable option to enforce data confidentiality while realizing deduplication. It encrypts/decrypts a data copy with a convergent key, which is derived by computing the cryptographic hash value of the content of the data copy itself. After key generation and data encryption, users retain the keys and send the ciphertext to the cloud. Since encryption is deterministic, identical data copies will generate the same convergent key and the same ciphertext. This allows the cloud to perform deduplication on the ciphertexts.

**Advantages:**

- The ciphertexts can only be decrypted by the corresponding data owners with their convergent key and master key.
- More efficient and secured.

## SYSTEM DESIGN

The proof of ownership [11] can be done using SHA-256. The convergent key encryption can be achieved by dekey using Secure Sharing Scheme whereas asymmetric encryption can be used for encrypting the files. The encrypted files can be decrypted only using both convergent key and master key provided by the user.

### SYSTEM ARCHITECTURE

The proof of ownership can be done using SHA-256. The convergent key encryption can be achieved by dekey [1] using Secure Sharing Scheme whereas asymmetric encryption can be used for encrypting the files.

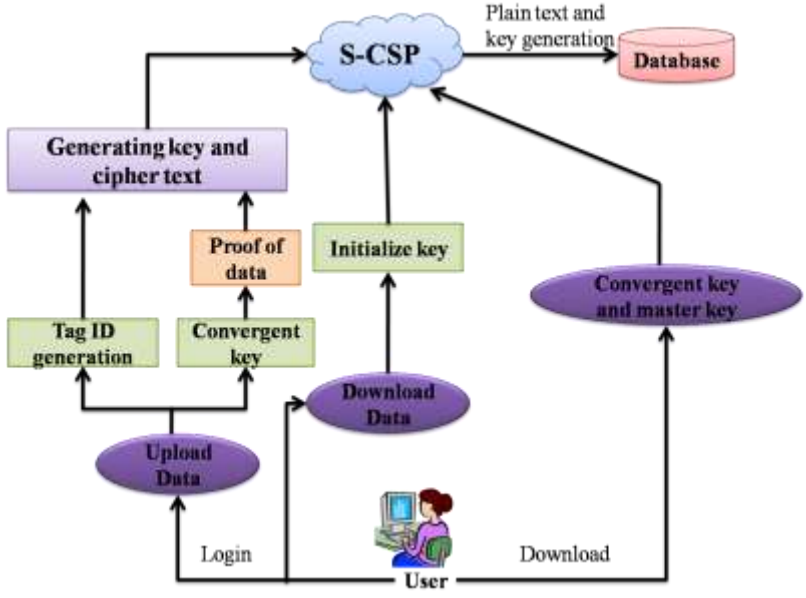


***Fig: 1 Proposed Architecture***

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

Dekey applies deduplication among convergent keys and distributes convergent key shares across multiple key, while semantic key algorithm to security of convergent keys and confidentiality of outsourced data. Main use of Dekey remains secures even the adversary controls a limited number of key servers. The encrypted files can be decrypted only using both convergent key and master key provided by the user.

**MODULE DESCRIPTION**

- Mastering File to Cloud Service Provider.
- Chunking the file chosen.
- Dekey Based Encryption.
- Hash value based Decryption.

**Mastering file to cloud service provider**

A user is an entity who wants to outsource data storage to the storage cloud service provider (S-CSP) and access the data later. User registers to the cloud server with necessary information and login cloud page for uploading the file. User chooses the file and uploads to server where the server store the file in rapid storage system and file level de-duplication is checked. We tag the file by using MD5 message-digest algorithm is cryptographic hash function [13] producing a 128-bit hash value typically expressed in text format as 32 digit hex value so that files of same are de-duplicated.

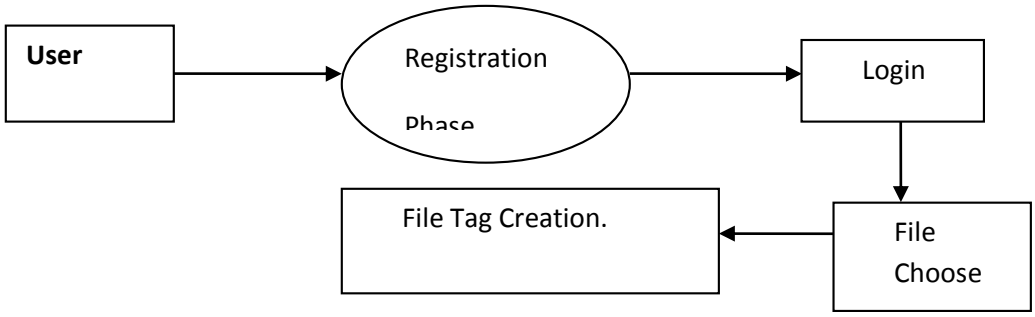


***Fig: 2 Mastering File to Cloud Service Provider***

**Chunking the file chosen**

Chunking the file chosen of fixed size and generating tags for each blocks chunked. After that generate convergent keys for each blocks split to verify block level deduplication [7]. Here we provide filename and password for file authorization in future. Encrypt the blocks by Triple Data Encryption Standard (3DES) algorithm. Here the plain text is encoded triple times with convergent key and so the while decoding the original content it also need the same key to decode again by triple times. Finally the original content is encrypted as cipher text and stored in Storage Cloud Service Provider (S-CSP) file storage system.

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

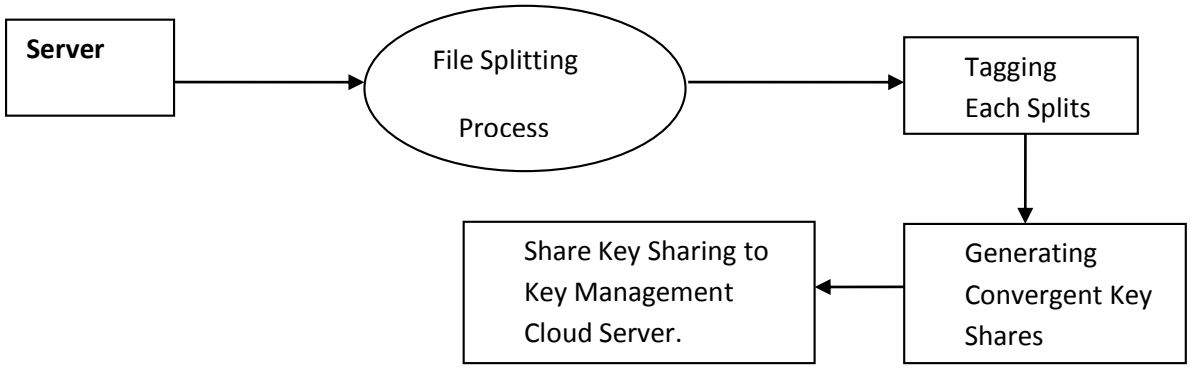


*Fig: 3 Chunking the file chosen*

**DE-KEY Based Encryption**

After encryption the convergent keys are securely shared with cloud service provider to Key Management Cloud Service Provider (KMCSP). Key management server checks duplicate copies of convergent keys in KMCSP. Key Management Server maintains Comma Separated Values (CSV) file to check proof of verification and store keys secure. The different users who share the common keys are referred by their own ownership. User request for deletion definitely need to prove proof of ownership to delete own contents.

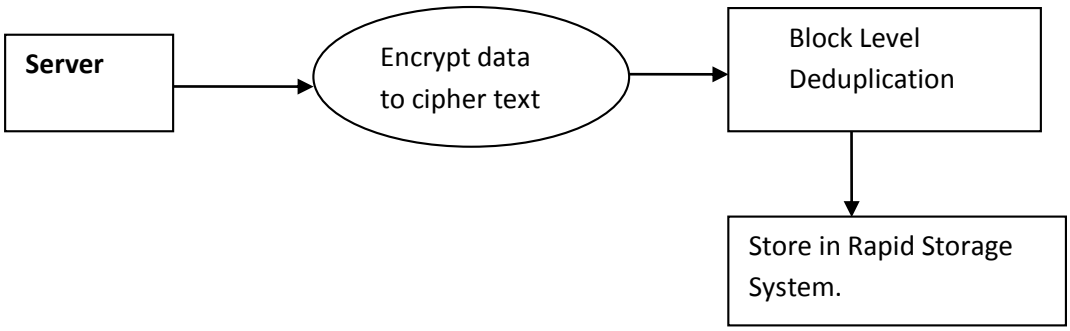


*Fig: 4 De-key Based Encryption*

**Hash value based decryption**

The final model where the user request for the downloading their own document which they have been upload and stored in cloud server. This download request needs proper ownership verification of the document here we create the ownership by unique tag [10] generated by MD5 algorithm and verifies existing tag of user. After verification the original content is decrypted by requesting the cloud server where cloud server request key management server for keys to decrypt and finally the original content is received by the user. The delete request will delete only the reference of the content shared by common users and not the whole content.

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

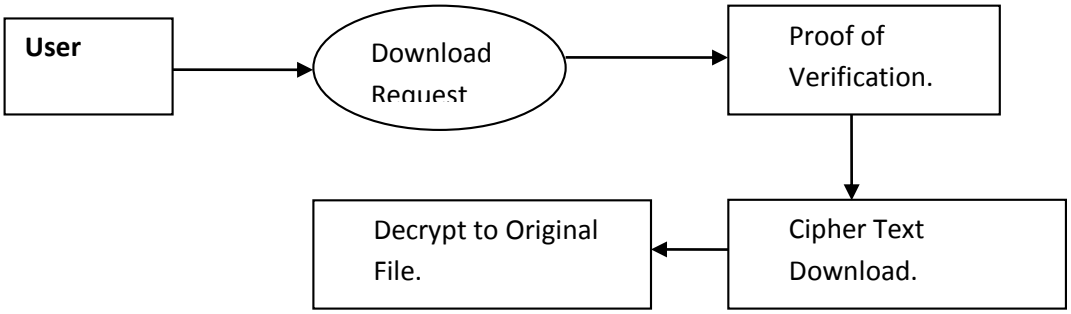


***Fig: 5 Hash value based decryption***

## SYSTEM REQUIREMENTS

**Software requirements**

- Windows operating system 7 and above.
- JDK 1.6.
- Tomcat 6.0.
- My Eclipse 6.8
- MySql 5.2.

**Hardware requirements**

- Hard Disk: 100 GB and Above.
- RAM: 2GB and Above.
- Processor: I3 and Above.

## CONCLUSION

In this paper, the duplicate copies of data in cloud storage are avoided. Data is secured through deduplication using convergent encryption. Through convergent encryption, convergent key is applied by deduplication using dekey. The user need not manage any master key because of convergent encryption. This helps in reducing the storage space in cloud. We demonstrate that the duplicate copies are not stored in cloud storage and it just has the reference to perform download operations through secure sharing scheme by implementing Dekey.

## REFERENCES

1. J. Li, X. Chen, M. Li, J. Li, P.P.C. Lee, and W. Lou, "Secure Deduplication with Efficient and Reliable Convergent Key Management", IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 6, pp. 1615-1625 , Jun 2014.
2. NIST's Policy on Hash Functions, Sept. 2012. [Online]. Available: http://csrc.nist.gov/groups/ST/hash/policy.html
3. AmazonCase Studies. [Online]. Available: https://aws.amazon. com/solutions/case-studies/#backup
4. P. Anderson and L. Zhang, ''Fast and Secure Laptop Backups with Encrypted De-Duplication,'' in Proc. USENIX LISA, 2010, pp. 1-8.
5. M. Bellare, S. Keelveedhi, and T. Ristenpart, ''Message-Locked Encryption and Secure Deduplication,'' in Proc. IACR Cryptology ePrint Archive, 2012, pp. 296-3122012:631.
6. G.R. Blakley and C. Meadows, ''Security of Ramp Schemes,'' in Proc. Adv. CRYPTO, vol. 196, Lecture Notes in Computer Science, G.R. Blakley and D. Chaum, Eds., 1985, pp. 242-268.
7. A.T. Clements, I. Ahmad, M. Vilayannur, and J. Li, ''Decentralized Deduplication in San Cluster File Systems,'' in Proc. USENIX ATC, 2009, p. 8.
8. J.R. Douceur, A. Adya, W.J. Bolosky, D. Simon, and M. Theimer, ''Reclaiming Space from Duplicate Files in a Serverless Distributed File System,'' in Proc. ICDCS, 2002, pp. 617-624.

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

9. J. Gantz and D. Reinsel, The Digital Universe in 2020: Big Data, Bigger Digital Shadows, Biggest Growth in the Far East, Dec. 2012. [Online]. Available: http://www.emc.com/collateral/analystreports/ idc-the-digital-universe-in-2020.pdf
10. R. Geambasu, T. Kohno, A. Levy, and H.M. Levy, ''Vanish: Increasing Data Privacy with Self-Destructing Data,'' in Proc. USENIX Security Symp., Aug. 2009, pp. 316-299.
11. S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, ''Proofs of Ownership in Remote Storage Systems,'' in Proc. ACM Conf. Comput. Commun. Security, Y. Chen, G. Danezis, and V. Shmatikov, Eds., 2011, pp. 491-500.
12. D. Harnik, B. Pinkas, and A. Shulman-Peleg, ''Side Channels in Cloud Services: Deduplication in Cloud Storage,'' IEEE Security Privacy, vol. 8, no. 6, pp. 40-47, Nov./Dec. 2010.
13. S. Kamara and K. Lauter, ''Cryptographic Cloud Storage,'' in Proc. Financial Cryptography: Workshop Real-Life Cryptograph. Protocols Standardization, 2010, pp. 136-149.
14. M. Li, ''On the Confidentiality of Information Dispersal Algorithms and their Erasure Codes,'' in Proc. CoRR, 2012, pp. 1-4abs/ 1206.4123. [16] D.T. Meyer and W.J. Bolosky, ''A Study of Practical Deduplication,'' in Proc. 9th USENIX Conf. FAST, 2011, pp. 1-13.
15. M. Mulazzani, S. Schrittwieser, M. Leithner, M. Huber, and E. Weippl, ''Dark Clouds on the Horizon: Using Cloud Storage as Attack Vector and Online Slack Space,'' in Proc. USENIX Security, 2011, p. 5.