IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-5, Issue-11 E-ISSN: 2347-2693

Client-Side Authorized Deduplication In Cloud Using PoW

P. Mounika[1*], S.Jyothsna[2]

[1*]CSE Dept., CVR College of Engineering, JNTU, Hyderabad, India
[2] IT Dept., CVR College of Engineering, JNTU, Hyderabad, India

[*]*Corresponding Author: pavushettymounika@gmail.com , Tel.:9989732910*

Available online at: www.ijcseonline.org

Received: 19/Oct/2017, Revised: 30/Oct/2017, Accepted: 22/Nov/2017, Published: 30/Nov/2017

***Abstract*—** Cloud computing is an effective and emerging technology for storing huge amounts of data. Most of the organizations and people are using cloud for storing various types of data. The critical challenge is to maintain the stored data without any redundancies due to billing nature of cloud. Deduplication is a popular technique used to remove duplicate copies from cloud. Existing deduplication techniques using convergent encryption does not support for authorized duplicate check. Authorized duplicate check is essential to protect the sensitivity and integrity of data that is stored. In this paper, the client-side authorized deduplication is implemented using hybrid cloud where the duplicate check is performed at client-side which improves data security and reduce network bandwidth. In this work, duplicate check for a file is performed by a token generated by private cloud based on privilege of user issued during system initialization and file content. Each file uploaded to the cloud is also bounded by a token to specify which kind of users is allowed to perform the duplicate check and access the files. The user is able to find a duplicate for this file if and only if there is a copy of this file and a matched privilege. To prevent unauthorized access, a secure proof of ownership (PoW) protocol is also implemented to provide the proof that the user indeed owns the same file instead of small information of file when a duplicate is found. It makes overhead to minimal compared to the normal convergent encryption and file upload operations.

***Keywords*—**Deduplication,AuthorizedDuplicateCheck**,**Confidentiality,ConvergentProtocol,HybridCloud,PoW

I. INTRODUCTION

The wide use of cloud has brought great convenience for data sharing and collection. The huge amounts of data from different organizations are being stored in cloud due to its popularity and ease. The data that is stored has to be maintained. Deduplication is a well-known and widely used technique to make data management scalable in cloud computing. Data deduplication is an emerging technology that introduces reduction of storage utilization and an efficient way of handling data replication in the backup environment. Deduplication occurs in four stages.

1.The incoming large data into small portions or 'chunks' using chunking method.
2. A unique identifier value is assigned to each chunk generated by using hashing algorithm.
3. The new incoming chunks are compared with the existing stored chunk using the unique identifier.
4. If the identifier matches, the redundant chunk will be removed and will be given the reference point; otherwise, new chunk will be stored.

Chunking can take place either at file level or block level. Deduplication can occur either at source or at the target. Target is the cloud and source may be the organization or user etc. Deduplication can be either client-side deduplication or target based deduplication. In client-side deduplication the deduplication occurs before the data is transferred into backup environment where as, in target based deduplication the deduplication occurs in backup environment.

Although data deduplication brings lot of benefits, privacy and security issues arise. As the data is stored in third party environment, the data is subjected to both insider and outsider attacks. Traditional encryption techniques although providing confidentiality will not support for deduplication. The users encrypt their data with their own keys which leads to different cipher text of same data. To overcome this, convergent encryption has been introduced which makes deduplication feasible and provides data confidentiality. In convergent encryption the key to encrypt data is obtained from the data by calculating cryptographic hash value of the data, identical data will generate same convergent key hence same cipher text. Existing deduplication techniques using convergent encryption[3] although providing confidentiality does not support for differential authorization duplicate check. This is important in many applications. In cloud computing environment, same file could be shared to many users. The user with certain

privileges should only be allowed for duplicate check and to download the file because an unauthorized person may delete or download the file to which he gain access. So, there is need to implement access control system.

The client side deduplication with convergent encryption is done in this work to provide confidentiality along with deduplication. In this system, deduplication is performed based on privilege of the user along with content of file. To perform authorized deduplication, the system of hybrid cloud is used. The privilege of the user is issued by the private cloud during system initialization. A file based deduplication is performed here. The source-based deduplication is performed to incur minimal overhead using client-side deduplication to reduce the burden on network, improve efficiency, security and reduce network bandwidth. The user is able to find a duplicate for the file if and only if there is a copy of this file and a matched privilege stored in cloud. We also implement a secure proof of ownership [4] protocol when a duplicate is found to provide the proof that user indeed owns the file when a duplicate is found to prevent from unauthorized access of file. If the proof is passed then subsequent users with the same file and privilege will be provided a pointer from the server without needing to upload the same file. A user can download the encrypted file with the pointer from the server, which can only be decrypted by the corresponding data owners with same privilege by their convergent keys.

Section II contain the related work on existing deduplication techniques and PoW, Section III contain the architecture, design goals ,description of proposed client-side deduplication technique using PoW and methodology with flow chart, Section IV describes implementation details, Section V evaluates the effect of file size using this technique and Section V concludes research work.

II. RELATED WORK

To provide security and deduplication Stores et al,Long et al.[8] proposed "Secure Data De-duplication", it provides both data security and space efficiency in storage and distributed storage systems. The keys to encrypt the data are generated from chunks, thus, similar chunks will result same cipher texts when encrypted. The encrypted chunk data cannot be used to deduce the key. The key to encrypt the data is known only to the user, even the system cannot reveal which chunks belongs to user. We have two models for deduplicated storage they are: authenticated and anonymous which provides security. These models make use of convergent encryption to provide security as well as to support deduplication. A unique key is used to encrypt the file. In the authenticated model, asymmetric key pairs are used to manage sharing of this key. In the anonymous model, a map reference for each authorized user is created and used for file sharing by sharing map offline.

Bellare et al, Keelveedhi et al, Ristenpart et al [9] proposed "Message Locked Encryption and Secure Deduplication", In this the key to encrypt and decrypt is derived from the message to achieve secure deduplication. Based on this, they made both practical and theoretical contributions. On the practical side, they used MLE schemes which included deployed schemes to provide ROM security. On the theoretical side, they used hash functions, deterministic encryption that are secure on correlated inputs and sample. Different assumption on different classes of message sources were extracted. It is inherently subject to brute-force attacks that can recover files falling into a known set. Ng et al, Wen et al, Zhu et al [10] proposed "Private Data Deduplication Protocols in Cloud Storage", This is first protocol for deduplication in private cloud. In this system the client holds the summary string of his private data. The client proves to the server that he is the owner of the data by using this summary string without revealing any further information. It can be viewed as the state-of-the-art public data Deduplication protocols of Halevi et al. The security of deduplication protocols for private data is formalized in the simulation based framework in the context of two-party computations. Based on cryptographic assumption private De-duplication protocols are constructed and analyzed. The proposed private data De-duplication protocol is secure that it can erasure code to fraction of bits in the presence of malicious adverseries by using erasure algorithms. Though it provide security the user's sensitive data are susceptible to both insider and outsider attacks.

Kamara et al, Lauter et al [12] proposed "Cryptographic cloud storage", in this the problem of constructing cloud storage over public cloud infrastructure is focused. The Public cloud is not completely trusted by the customer. Several architectures based on cryptographic primitives were proposed to solve the problem. The benefits provided by architecture for both customer and service provider are reviewed, and advance in cryptography for cloud storage are discussed. Meyer et al, barsky et al [13] proposed "A Study of Practical De-duplication", In this practical study they collected data from 857 desktop computers at Microsoft for 4weeks. They analyzed the data to determine data deduplication relative efficiency considering whole-file versus block-level elimination of redundancy. It found that file-level De-duplication achieves about three quarters of the space savings than block-level elimination to store live file systems and 87% of the savings for backup images. File fragmentation finding that it is not prevalent, and updated prior file system metadata studies, finding that the distribution of file sizes continues to skew toward very large unstructured files.

J. Li, X.Chen, M.Li, J.Li, P.Lee, and W. Lou[15], proposed "Secure Deduplication with Efficient and Reliable Convergent Key Management". This paper addresses on management of keys. Although convergent encryption provides secure deduplication, management of huge number of convergent keys effectively and reliabilly is critical challenge. To address the problem effective key management they proposed baseline approach where each user holds a master key independently for encrypting convergent keys and storing them in cloud storage. However, this generates enormous number of keys with increasing number of users. For this they proposed a concept of Dekey, by which users need not manage any keys instead distribute the convergent keys securely over multiple servers. This is effective and secure but increases network bandwidth. This Deduplication systems cannot support differential authorization duplicate check.

Halevi et al, Harnik et al, Pinkas et al [4] proposed "Proofs of Ownership in Remote Storage Systems" This is used to identify attacks in client-side deduplication. An attacker can gain access to files by using small signature of files that is hash value generated from file content. An attacker can convince the storage server by using this hash signature that he owns the file and can download the file. As in most of the storage servers the deduplication check is done by comparing hash value of the content of file instead of the file. To overcome such attacks, a secure proof of ownership(PoWs), is introduced. This is used to prove the server that client holds the entire file rather than short information of file. They formalize the concept of proof-of-ownership, under various security definitions and efficiency requirements of Petabyte scale storage systems.

In this paper, we use file based deduplication scheme. We perform source-based deduplication to incur minimal overhead using client-side deduplication to reduce the burden on network, improve efficiency, security and reduce network bandwidth. A secure PoW scheme is implemented to prevent the file access from attackers. A concept of hybrid cloud with authorized deduplication is implemented to provide higher security.

III. METHODOLOGY

A. *Secure Deduplication using Hybrid Architecture:*

There are three entities defined in our system as shown in figure 1, those are

- Users
- Private cloud
- S-CSP in public cloud

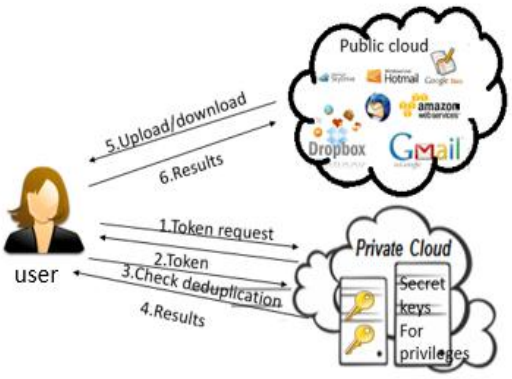

Figure 1. System architecture.

- *S-CSP:* This is an entity that provides a data storage service in public cloud. The S-CSP provides the data outsourcing service and stores data on behalf of the users. To reduce the storage cost, the S-CSP eliminates the storage of redundant data via de-duplication and keeps only unique data by performing client-side authorized deduplication using private cloud. In this paper, we assume that S-CSP is always online and has abundant storage capacity and computation power.

- *Data Users:* A user is an entity that wants to outsource data storage to the S-CSP and access the data later. In a storage system supporting deduplication, the user only uploads unique data but does not upload any duplicate data to save the upload bandwidth, which may be owned by the same user or different users. In the authorized de-duplication system, each user is issued a set of privileges in the setup of the system. Each file is protected with the convergent encryption key and privilege keys to realize the authorized de-duplication with differential privileges.

- *Private Cloud:* Compared with the traditional data deduplication architecture in cloud computing, this is a new entity introduced for facilitating user's secure usage of cloud service. Specifically, since the computing resources at data user/owner side are restricted and the public cloud is not fully trusted in practice, private cloud is able to provide data user/owner with an execution environment and infrastructure working as an interface between user and the public cloud. The private keys for the privileges are managed by the private cloud, who answers the file token requests from the users. The duplicate check for files is also done by the private cloud when user makes an upload request with the token if there is no duplicate we can directly upload to cloud, if duplicate is found user has to run proof of ownership protocol, if user proves the ownership that he has the file then, a link to the file stored in cloud is provided. The interface offered by the private cloud allows user to submit

files and queries to be securely stored and computed respectively.

Hybrid clouds generally have twin clouds (private cloud and public cloud). This architecture is used for data de-duplication. For example, an enterprise might use a public cloud service, such as Amazon S3, for archived data, but continue to maintain in-house storage for operational customer data. Alternatively, the trusted private cloud could be a cluster of virtualized cryptographic co-processors, which are offered as a service by a third party and provide the necessary hardware based security features to implement a remote execution environment trusted by the users.

*B. Design goals***:**

In this paper, we propose a new source-based client-side deduplication technique which supports for differential authorized duplicate check. The objective of this paper is to perform differential authorization, authorized duplicate check and to obtain security and confidentiality.

Differential authorization: Every user should be assigned with a privilege to generate token for a file, so that they can perform deduplication based on their privilege. Token should not be generated without a valid privilege. The private cloud maintains the privilege of the users and generates token when request is made by the user.

Authorized duplicate check: The user can check for duplication of a file with the help of token generated by the private cloud by using their individual private keys based on their privilege. we perform client-side deduplication to reduce network bandwidth, to increase efficiency and security. Private cloud performs duplicate check directly and tells the user if there is any duplicate.

Security: Unauthorized users without appropriate privileges or file should be prevented from getting or generating the file tokens for duplicate check of any file stored at the S-CSP. The users are not allowed to collude with the public cloud server to break the unforgeability of file tokens. It requires that any user without querying the private cloud server for some file token, he cannot get any useful information from the token, which includes the file information or the privilege information. In our system, the private cloud is honest but curious and will honestly perform the duplicate check upon receiving the duplicate request from users.

Confidentiality: The confidentiality of data has to be maintained. Unauthorized users without specified privilege should be prevented from accessing the data stored in public cloud. The users including public and private cloud servers should not get any information regarding original data from the encrypted data. A higher level confidentiality should be achieved compared to existing confidentiality techniques using convergent encryption. In proposed system we perform convergent encryption based on privilege and content of file using AES algorithm so, a higher level confidentiality is achieved.

Ownership verification protocol: The subsequent ownership protocol has to be implemented to prevent unauthorized access of file when a duplicate is found. An unauthorized user may gain the token by eavesdropping and claim for the file. This has to prevented by a effective ownership algorithm. In this paper, we propose an algorithm that runs between user who wants to upload file and the storage server. This algorithm generates a value called as POW token from data of file. The values generated by both user and storage server are compared and token is verified. The user cannot prove the ownership if he has small information of file.

C. Algorithm

Algorithm to generate token to verify ownership of file

Input: file.

Output: token (hash value).

Step1: divide the file size into n parts.

N1= File size/n;

Step 2: Divide the file into blocks of size N1 and store as f0, f1…fn.

Step 3: calculate hash value for each block.

```
FOR i=0 to n

Token(i)=sha1(fi);

End FOR
```

Step 4: Calculate hash value of cumulative blocks until we get single value.

```
WHILE n not equal to zero

FOR i= 0 to n and j=0 to n/2

Token(j)=Sha1(Token(i)+Token(i+1));

i=i+2;

j=j+1;

END FOR
```


n=n/2;

END WHILE

Step 5: token= token (0)

Output the value to cloud for verification

D. Working of PoW Algorithm.

It generates a token to prove the ownership of file when a duplicate is found. The confidentiality is maintained as we are going to calculate the PoW token for an encrypted file. The users prove the ownership if and if he has the entire file. PoW is calculated by dividing the entire file into chunks **a**nd calculating hash value for each chunk. Then calculates hash value for cumulative chunks until it gets the single hash value which is called as PoW token by which we verify the ownership of file.

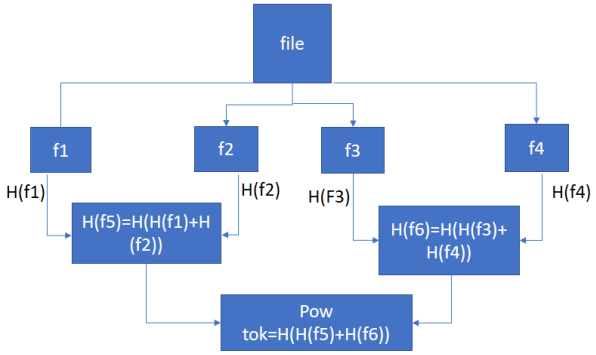

Figure 2. PoW Architecture

E. Modules Description:

The detailed architecture of the design is showed in figure1. We can get the processing details from this architecture. Six different types of modules are present in the architecture. Tag Generator Module, Token Generator Module, Encryption and Decryption Module, Duplicate Checker Module, PoW Module, File Storager Module. User login details are required to upload or download a file and the details of modules mentioned below:

Tag Generator Module:
a) Generates a tag for a file.
b) Requests for token.

$$Tag= H(F) \quad (1)$$

Token Generator Module:
a) Verifies user identity.
b) Generates token for deduplication.
c) Maintains and assigns privileges to a user.

Here the privilege is the role assigned to users during registration by private cloud. Different users have different access permissions based on their privilege. Every privilege has a secret key maintained in this module to provide confidentiality. Based on the secret key and tag it generates token.One cannot generate a token by using others privilege as this module maintains the information of users privilege and a secrect key is maintained for each privilege.

TABLE 1 Privilege Table

Sno	Privileges	Secret key
1	Developer	Sk1
2	Administrator	Sk2
3	Team Leader	Sk3

$$Token=H(H(F)+Ski) \quad (2)$$

Encryption and Decryption Module:

a) Encrypts the file using token and send request for upload.
b) Decrypts the files.

Duplicate Checker Module:
a) It maintains a map between existing files and associated token.
b) Check for deduplication.

PoW Module
a) Calculates POW token.
b) Verifies Proof of Ownership of the file.

In this module it generates a token to prove the ownership of file when a duplicate is found. The confidentiality is maintained as we are going to calculate the POW token for an encrypted file.

File Storager Module:
a) Uploads the file to cloud server.
b) Downloads the file from cloud server.
c) Delete the files from cloud server.

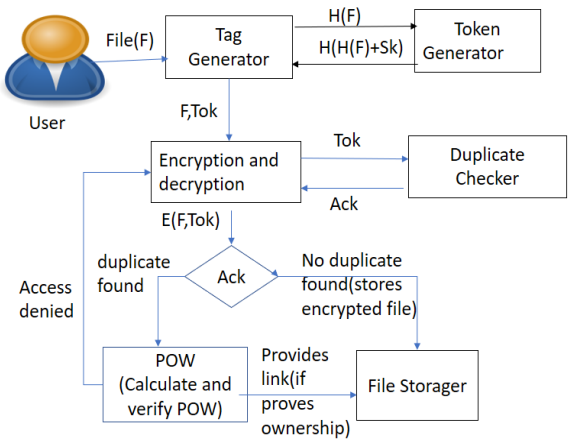

Figure 3. Flow Diagram

IV. RESULTS AND DISCUSSION

We implement a prototype of the proposed source based client-side authorized deduplication system, in which deduplication check takes place in private cloud and stores data in public cloud. We model three entities as separate java programs. A Client program is used to model the data users to carry out the file upload process. A Private Server program is used to model the private cloud which manages the private keys and handles the file token computation and perform deduplication. A Storage Server program is used to model the S-CSP which stores files. Here we are using dropbox as public cloud. In order to perform operations on dropbox we have to create an app in app console, this gives an app key and secret key to perform operations on dropbox.

Our implementation of the Client provides the following function calls to support tag generation and encryption along the file upload process.

- *FileTag(File)*—It computes SHA-1 hash of the File as File Tag;
- *TokenReq(Tag,UserID)*—It requests the private cloud for file token generation with the file tag and User ID.
- *DupCheckReq(Token)*—It requests the private server for duplicate check of the file by sending the file token received from private cloud.
- *FileEncrypt(File)*—It encrypts the file with convergent encryption using AES algorithm, where the convergent key is from SHA-1 Hashing of the file.
- *PoWgeneration(encryptedfile)*- It generates PoW token to verify ownership of file when a duplicate is found.
- *FileUpload(FileID, File, Token)*—It uploads the file to the storage server if the file is Unique and updates the file token in private cloud.
- *FileDownload(FileID,Token)*- It downloads the file from storage if the file token is valid.

Our implementation of the Private cloud includes corresponding request handlers for the token generation and maintains a key storage with Hash Map and also provides deduplication by maintaining a map between existing files and associated token.

- *TokenGen(Tag, UserID)*—It loads the associated privilege keys of the user and generate the token with HMAC-SHA-1 algorithm.
- *DupCheck(Token)*—It searches the File to Token Map for Duplicate.
- *Verifyownership(PoWToken*): It verifies the ownehip of a file when a duplicate is found.

Our implementation of the Storage Server provides data storage.

- FileStoreger(FileID, File, Token)—It stores the File on Disk and updates the Mapping in private cloud.

In client-side authorized deduplication the users with same privilege and file are provided with a link to the file in the storage through which they can access the file if they prove the ownership. If there is a same file with different user privileges then then we store the same file as two different copies in the storage in this way deduplication takes place.

V. EVALUATION

The evaluation focuses on overhead caused by token generation of file, encryption of file, deduplication and file upload process. The overhead is evaluated by different factors such as file size etc. We break down the upload process into six steps, 1) Tagging, 2) Token Generation, 3) Duplicate Check, 4) PoW Verification, 5) Encryption, 6) Transfer. For each step, we record the start and end time of it and therefore obtain the breakdown of the total time spent.

File size: Effect of file size to time spent on different steps is evaluated by uploading 50 unique files of different file sizes and record the time break down. The average time for different file sizes and the steps is plotted. The time increases linearly for tagging, PoW verification, encryption and file transfer with increasing file size. The time taken for token

generation and duplicate check remains constant for all file sizes because we use file metadata for computation in these steps. With increasing file size the overhead increases.

VI. CONCLUSION

Data deduplication in cloud reduces the storage overhead created by the huge amount of data by not letting the duplicate copies to be stored on the cloud. Source based client-side deduplication using private cloud is performed here to reduce size, to improve network bandwidth, efficiency and security. The authorized data deduplication was proposed to protect the data security by including differential privileges of users in the duplicate check. This system supports authorized duplicate check in hybrid cloud architecture for addressing fault tolerance, in which the duplicate-check tokens of files are generated and deduplication is done by the private cloud server with private keys and tokens. Convergent encryption is used here to enhance the confidentiality which overcomes the drawbacks of traditional encryption. Proposed system includes proof of data ownership protocol along with deduplication checker which helps to implement better security issues in cloud computing from both insider and outsider attacks. We showed that our authorized duplicate check scheme incurs minimal overhead compared to existing deduplication techniques.

REFERENCES

[1] Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick P. C. Lee, Wenjing Lou" *A Hybrid Cloud Approach for Secure Authorized De-duplication*" in vol: pp no-99, IEEE, 2014.

[2] S. Quinlan and S. Dorward, "*Venti: A new approach to archival storage*," in Proc. 1st USENIX Conf. File Storage Technol., Jan. 2002, p. 7.

[3] 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. Int. Conf. Distrib. Comput. Syst., 2002, pp. 617–624.

[4] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "*Proofs of ownership in remote storage systems*," in Proc. ACM Conf. Comput. Commun. Security, 2011, pp. 491–500

[5] D. Ferraiolo and R. Kuhn, "*Role-based access controls*," in Proc. 15th NIST-NCSC Nat. Comput. Security Conf., 1992, pp. 554–563.

[6] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "*Role-based access control models*," IEEE Comput., vol. 29, no. 2, pp. 38–47, Feb. 1996.

[7] A. Shamir, *How to Share a Secret*, Communications of the ACM, vol. 22, no 11, pp. 612-613, 1979.

[8] M.W. Storer, K. Greenan, D.D.E. Long, and E.L. Miller, "*Secure Data De-duplication*", Proceedings of the 4th ACM international workshop on Storage security and survivability, pp1-10, 2008

[9] M. Bellare, S. Keelveedhi, and T. Ristenpart, "*Message-locked encryption and secure deduplication*," in Proc. 32nd Annu. Int. Conf. Theory Appl. Cryptographic Techn., 2013, pp. 296–312.

[10] W.K. Ng, Y. Wen, and H. Zhu, *Private Data De-duplication Protocols in Cloud Storage*, Proceedings of the 27th Annual ACM Symposium on Applied Computing,S. Ossowski and P. Lecca, Eds.,pp. 441-446, 2012.

[11] R.D. Pietro and A. Sorniotti, *Boosting Efficiency and Security in Proof of Ownership for De-duplication*,in Proceedings of ACM Symposium on Information, Computer and Communication Security, H.Y. Youm and Y. Won, Eds., pp. 81-82, 2012.

[12] S. Kamara and K. Lauter, *Cryptographic Cloud Storage, in Proceedings of Financial Cryptography: Workshop on Real-Life Cryptograph*. Protocols Standardization, pp.136-149, 2010

[13] D.T. Meyer and W.J. Bolosky, *A Study of Practical De-duplication,* in Proceedings of 9th USENIX Conference on File and Storage Technologies, pp. 1-13, 2011.

[14] M. Bellare, S. Keelveedhi, and T. Ristenpart, "*Dupless: Serveraided encryption for deduplicated storage*," in Proc. 22nd USENIX Conf. Sec. Symp., 2013, pp. 179–194.

[15] J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou, *"Secure deduplication with efficient and reliable convergent key management*," in Proc. IEEE Trans. Parallel Distrib. Syst., http://doi.ieeecomputersociety.org/10.1109/TPDS.2013.284, 2013.

[16] C. Ng and P. Lee, "*Revdedup: A reverse deduplication storage system optimized for reads to latest backups*," in Proc. 4th AsiaPacific Workshop Syst., http://doi.acm.org/10.1145/2500727. 2500731, Apr. 2013.

[17] V.P.Muthukumar and R.Saranya, "*A Survey on Security Threats and Attacks in Cloud Computing*", International Journal of Computer Sciences and Engineering, Page No : 120-125, Volume-02 , Issue-11, E-ISSN: 2347-2693, Nov - 2014

[18] J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl, "*A secure data deduplication scheme for cloud storage*," Tech. Rep. IBM Research, Zurich, ZUR 1308-022, 2013.

[19] M. Bellare, C. Namprempre, and G. Neven, "*Security proofs for identity-based identification and signature schemes*," J. Cryptol., vol. 22, no. 1, pp. 1–61, 2009.

Authors Profile

Ms. P. Mounika pursued Bachelor of Technology from Bhoj Reddy Engineering college for Women, Hyderabad in 2015 and Master of Technology from CVR College of Engineering, Hyderabad in 2017. Her main research work focuses on Cloud computing, Network Security.

Mrs Jyothsna Sundargiri pursued M.C.A from Kakathiya University, Warangal in 2001 and Master of Technology from J.N.T.U, Hyderabad in year 2010. She is currently Pursuing Ph.D and working as Assistant Professor in Department of Computer Science and Engineering in CVR College of Engineering,Hyderabad since 2011. She has 13 years of teaching experience. Her main research work focuses on Cloud computing, Network Security and Data Analytics.

