A Secured and Authorized Data Deduplication in Hybrid Cloud with Public Auditing

Sharma Bharat
P.G. Student,
SSVPS's BS Deore College of Engineering,
Dhule, 424005,
India

Mandre B.R.
Associate Professor,
SSVPS's BS Deore College of Engineering,
Dhule, 424005,
India

ABSTRACT
The popularity and widespread use of Cloud have brought great convenience for data sharing and data storage. The data sharing with a large number of participants take into account issuers like data integrity, efficiency and privacy of the owner for data. In cloud storage services one critical challenge is to manage ever-increasing volume of data storage in cloud. To make data management more scalable in cloud computing field, deduplication a well-known technique of data compression to eliminating duplicate copies of repeating data in storage over a cloud. Even if data deduplication brings a lot of benefits in security and privacy concerns arise as user's sensitive data are susceptible to both attacks insider and outsider. A convergent encryption method enforces data confidentiality while making deduplication feasible. Traditional deduplication systems based on convergent encryption even though provide confidentiality but do not support the duplicate check on basis of differential privileges. This paper presents, the idea of authorized data deduplication proposed to protect data security by including differential privileges of users in the duplicate check. Deduplication systems, users with differential privileges are further considered in duplicate check besides the data itself. To support stronger security the files are encrypted with differential privilege keys. Users are only allowed to perform the duplicate check for files marked with the corresponding privileges to access. The user can verify his/her presence of file after deduplication in cloud with the help of a third party auditor by auditing the data. Further auditor audits and verifies the uploaded file on time. Therefore, this paper creates benefits to both the storage provider and user by deduplication technique and auditing technique respectively.

General Terms
Deduplication, hybrid cloud, Cloud security.

Keywords
Authorized check duplicates, confidentially, auditing.

1. INTRODUCTION
Hiding platform and implementation details unlimited virtualized resources provided to the users as a service is a cloud computing. Presently cloud service provided to the users offered high available storage and massively parallel computing of resources at relatively low costs. But the question is about the cloud users with different privileges store data on cloud is a most challenge issue in managing cloud data storage system [7].

Deduplication is technique which make data manage more scalable in cloud computing [2]. Data deduplication defines as a data compression technique which eliminates duplicate copies of repeat data in storage space. This technique is use to improve storage utilization and also apply to reduce the number of bytes that must be sent before upload in data transfers. Instead to keep same content data copies multiple times deduplication remove repetitive data and keep only one physical copy while refer other respective redundant data to that copy [3].

Deduplication can be applied to data which are in primary storage, cloud storage, backup storage for replication transfers[1]. The ways from which deduplication method execute, many types of there are, but basically only 3 types are in consideration which are as ideal deduplication process type as block level, file level and byte level by the names itself deduplicate process worked respectively on that content.

Even if data deduplication has lot of foredeal, but user with sensitive data are worried about both outsider/insider attacks. So deduplication of data must be handle security and privacy. But with traditional encryption different users encrypt data with their own key, which makes likeness data with different user key makes different ciphertext for that data which is unable for deduplication. The convergent encryption allows encrypt/decrypt data with convergent key on the data thus makes possible to apply to check duplicates [3]. Thus with uploading user's data as ciphertext to cloud resolved security issues. To prevent form unauthorized access, proofs of ownership protocol can be used as privacy constraint [4]. Proof of ownership; user can download the decrypted and obtain particular data with convergent keys by specifying its ownership. Thus by using convergent encryption and proof of ownership both security and privacy issues resolve.

Still the system can't work on privilege level field, means user upload file with some set of privileges on its and on the basis on convergent encryption doesn't provide any deduplication on it [1]. Thus not support duplicate check with different privileges set provided by the data owner.

This paper leads to removes all those problems by considering hybrid cloud architecture, in which public cloud makes available to data owner for providing storage place which will managed by private cloud act as an proxy to allow data owner and user with security and privacy along with different privileges set.

1.1 Contribution
As the data is uploaded in public cloud even if it is in encrypted from for more privacy purpose this paper contributed by applying Public Auditing to the file uploaded in public cloud. A new Deduplication schema with support both security and privacy with different privilege set provided by data owner and also includes auditing. Auditing is a process in which after

uploading file by data owner an unique auditor will audit the respective file and make metadata of it's by assigning the unique audit ID number which will act as TPA.

Finally, we are implement a prototype of Authorized Data deduplication along with auditing facility involved in it over a hybrid network.

2. NOTATION AND PRELIMINARIES

Table 1. Notation

Acronym	Description
(pk_U, sk_U)	User's public and secret key pair
k_F	Convergent encryption key for file F
P_U	P_U Privilege set of a user U
P_F	Specified privilege set of a file F
$\emptyset'_{F,p}$	Token of file F with privilege p
TPA	Third Party Auditor

2.1 Symmetric encryption

Symmetric encryption uses a common secret key k used to encrypt and decrypt. A symmetric encryption consists of three primitive functions:

- $GenKey_{SE}(1^{\lambda}) \rightarrow k$ is key generation algorithm to generates k with use of security parameter 1^{λ};
- $DEnc_{SE}(k, M) \rightarrow C$ is symmetric encryption algorithm, takes secret k and message M and then outputs ciphertext C;
- $DDec_{SE}(k, C) \rightarrow M$ is symmetric decryption algorithm, takes secret k and ciphertext C and then outputs original message M.

2.2 Convergent encryption

Convergent encryption [3], [5] provides data confidentiality for deduplication process. A data owner or user computes a convergent key by system and encrypts the original data with the convergent key. User derives a *tag* for data copy; this tag will be used to detect duplicates. By considering the tag correctness property holds means, if two data copies are same, then their respective tags are also same. For detecting duplicates, user first sends tag to server side to check whether the identical copies of the data are already present in storage or not. Convergent key and tag both are independently derived so, the tag cannot be used to deduce convergent key along with compromise data confidentiality. Server will store encrypted data copy and its corresponding tag. Convergent encryption scheme defines with four primitive functions as:

- $KeyGen_{CE}\ (M) \longrightarrow K$ is key generation algorithm, which maps data copy M to a convergent key K;
- $Enc_{CE}(K, M) \longrightarrow C$ is symmetric encryption algorithm, takes both the convergent key K and data copy M as inputs and then outputs a ciphertext K;
- $Dec_{CE}(K, C) \longrightarrow M$ is decryption algorithm, takes both the ciphertext C and the convergent key K as inputs and then outputs the original data copy M; and
- $TagGen(M) \longrightarrow T(M)$ is generation algorithm for tag that maps the original data copy M and outputs a tag $T(M)$.

2.3 Proof of ownership

PoW enable users to access data copies storage in server by proving their ownership for it. PoW is interactive algorithm which runs by a prover (i.e., user) and a verifier (i.e., storage server) [4]. Verifier derives a short value $\phi(M)$ from a data copy M. Ownership of the data copy M, prove by the prover with sending ϕ' to the verifier such that $\phi' = \phi(M)$.

2.4 Identification Protocol

An identification protocol II has Proof and Verify this two phase. In the first phase of Proof, a user or prover U can demonstrate his/her own identity to a verifier by performs some identification proof related to his/her identity. The input of the user or prover is his/her private key sk_U which is sensitive information as private key of a public key in his/her certificate or any credit card number etc. that he/she would not share with the other users as to maintain privacy. Now the verifiers perform verification with input of public information pk_U related to sk_U. the result of protocol, is that the verifier outputs either accept or reject for proof is passed or fail [9].

2.5 MAC-based Solution

There are two ways to make use of MAC [8] for authenticate of given data. In trifling, upload data blocks with their MACs to the server. Now by sends the corresponding secret key s_k to the TPA. After that TPA can randomly retrieve blocks with their MACs and check the correctness via s_k. Apart from the high (linear in the sampled data size) communication and computation complexities, the TPA requires the knowledge about the data blocks for verification process.

3. SYSTEM MODEL

Cloud User: A cloud user is which who wants to outsource data on public storage which acts as a public cloud in cloud computing. A system provides authenticate used to enter in system upload data with particular set of privileges for further accessing the uploaded data to download.

Public Storage: Public Storage is an storage disk which allow to store the users data on its with include of authorized and not allow to upload the duplicate data. Thus save storage space and bandwidth of transmission. This uploaded data is in encrypted form, only a user with respective key can decrypt it.

Private Cloud: A private cloud acts as a proxy to allow both data owner and user to securely perform duplicate check along with differential privileges.

Auditor: Auditor is a TPA work as expertise and capabilities where cloud users do not have to trust to assess the cloud storage service reliability on behalf of the user upon request.

The set of privileges and the symmetric key for each privilege is assigned and stored in private cloud. The user registers into the system, privileges are assigned to user according to identity given by the user at registration time; means on basis of post which access by the user. The data owner with a privilege issued set wants to upload and share a file to users, further the data owner performs identification and sends the file tag to the private server. Private cloud server verifies the data owner and computes the file token and will send back the token to the data owner.

The data owner sends this file token and a request to upload a file to the storage provider. If duplicate file is found then user needs to run the PoW protocol with the storage provider to prove he/she has an ownership of respective file. In the PoW result; if proof of ownership of file is passed then user will be provided a pointer for that file. And on the second case; for no

duplicate is found for the file, the storage provider will be return an signature for the result of that proof for the particular file. To upload file user sends the privilege set as well as the proof to the private cloud server as a request. The private cloud server verifies the signature first on receiving the request for the user to upload file. If the result of signature verification is passed, private cloud will compute the file token with each privileges from the privilege set given by the user, which will be returned to the user.

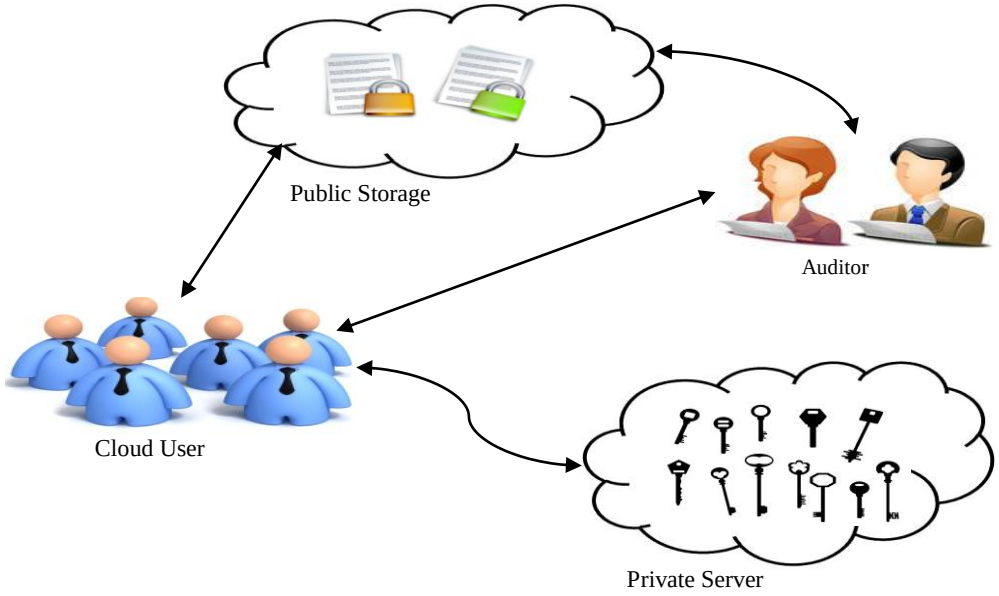

Figure 1. System Model of Authorized Deduplication

Finally user computes the encryption. User encrypts the file with a key k and the key k is encrypted into ciphertext with each key in the file token given by the private cloud server. Then the user uploads the encrypted file, file tag, encrypted key. Suppose user wants to download file F. The user first uses his key to decrypt the encrypted key and obtain key k. Then the user uses k to recover the original file F.

The user may or may not be sure about the presence of file in the cloud. Therefore, for user benefits an auditing scheme is used to audit the files stored in the public storage. User selects an auditor from the cloud and sends the metadata about the files going to upload in cloud to the auditor. Auditor issues audit message or challenge to the public storage to make sure that cloud server had retained the data file F properly at the time of the audit. Public cloud storage will derive a response message from a function of the stored data file F and its verification metadata by execute $GenProof$. The TPA then verifies the response via $VerifyProof$ for that particular users data file.

3.1 Design Goals

To develop a system the three goals must be archive by system. They are privacy preserving, security and auditing. With privacy preserving the system must possess Differential authorization in which based on privilege level an authorized user able to get file token. And Authorized duplication check for a certain file will be check by public with only issued token by private on basis of private keys and privileges of the authorized user. The goal related to security of file token are unforgeability which will assure private server issued to user request and indistinguishability token doesn't give an useful information to one another. To archive data confidentiality convergent encryption with higher level of privileges can be handling. To verify the user correctness for its data auditing help us out in our system.

4. AUTHORIZED DEDUPLICATION WITH AUDITING

4.1 Main Idea

To support deduplication with authorized, the tag of a file F will be determined by the file F and the privilege. As traditional notation of file is tag so to show the difference file token for simplicity. For supporting authorized access for user, a secret key k_p will be bounded with a privilege p to generate file token for it. Now consider $\phi'_{F,p} = TagGen(F, k_p)$ denote the token of F that is only allowed to access by user with privilege p defined by the users itself. In another word, the token $\phi'_{F,p}$ could only computed by the users with privilege p. Result view as, if file uploaded by user with duplicate token $\phi'_{F,p}$, then duplicate check sent from another user will be successful if and only if he/she also had the file F and privilege p. This token generation function could be easily implement as $H(F, k_p)$, where denotes as $H(.)$ a cryptographic hash function.

There are N users in the system and the privileges in the universe are defined as $P = \{p_1, \dots, p_s\}$. for each privilege p in P for set, private key k_p will be selected. For user U with a set of privileges P_U, he/she will assign the set of keys as $\{k_{p_i}\}_{p_i \epsilon P_U}$.

Binary relations define as $\mathbb{R} = \{(p, p')\}$. Along with given two privileges p and p'. And the value p matches p' if and only if $\mathbb{R}(p, p') = 1$. Such generic binary relation definition could be represented as based on the background of applications which include a common concept in relation of hierarchical system. More exactly, hierarchical relation is which, p matches p', only when p a higher-level privilege compare with p'.

The target file space underlying given ciphertext C is drawn from a message space $S = \{F_1, \dots, F_n\}$ of size n, the public cloud server can recover F after at most n off-line encryptions. That is,

for each $i = 1, \dots, n$, it simply encrypts F_i to get a ciphertext denoted by C_i. if $C = C_i$, it means that the underlying file is F_i. The security constraints are possible only when such a message is unpredictable, but traditional convergent encryption will be insecure for predictable file.

We design and implement new system which could protect security for predictable message. The *main idea* of our technique is that novel encryption key generation algorithm. To define tag generation functions and convergent keys, we will use hash functions in this section. To support duplicate check in traditional convergent encryption the key is derived from the file F by using some cryptographic hash function $k_F = H(F)$. To avoid the deterministic key generation process, encrypted key k_F of file F in our system will be generated with aid of private key cloud server with privilege key k_p. Encryption key can be sceneries as form $k_{F,p} = H_0\big(H(F), k_p\big) \oplus H_2(F)$, where H_0, H and H_2 all are defined as cryptographic hash functions used in system. The file F is encrypted with another key k, while k will be encrypted with $k_{F,p}$. This way, both private cloud server and public storage cannot decrypt the ciphertext. Moreover, on bases of the security of symmetric encryption makes secure to the public storage. For public storage, if the file is unpredictable from, then it is secure also. The details of the scheme, for simplicity which has been instantiated with hash functions are described below.

4.2 System Setup

The privilege universe P and the symmetric key k_{p_i} for each $p_i \epsilon\, P$ will be selected for private cloud as above. Identification protocol $\Pi = (\text{Proof}, \text{Verify})$ is also defined. The proof of ownership POW is instantiated by H, H_0, H_1 and H_2 a hash functions which will be shown as follows. Private cloud maintains each user's identity and its corresponding privilege stored in a form of table.

4.3 File Uploading

Suppose that a data owner with privilege p wants to upload and share a file F with users whose privilege belongs to the set $P = \{p_j\}$. Data owner performs the identification and sends $H(F)$ to the private cloud server. Two file tag sets $\{\phi_{F,p_\tau} = H_0\big(H(F), k_{p_\tau}\big)\}$ and $\{\phi'_{F,p_\tau} = H_1\big(H(F), k_{p_\tau}\big)\}$ for all p_τ satisfying $\mathbb{R}(p, p_\tau) = 1$ and $p \epsilon P_U$ will be sent back to the user if the identification process passes. Then after receiving the tag $\{\phi_{F,p_\tau}\}$ and $\{\phi'_{F,p_\tau}\}$, the user will interact and send these two tag sets to the public storage. If a file duplicate is found, the user needs to run the PoW protocol POW with the public storage to prove the file ownership. If the result of POW passed, the user will be now allow to provided a pointer of the respective file. On the author side, if no duplicate is found in cloud storage, a POW from the public storage will be returned to the user which could be a signature. Then user sends the set privilege $P = \{p_j\}$ as well as the proof to the private cloud server for upload request of the file. After receiving request private cloud server verifies signature first as purpose of generic access. If the result of its passed, private cloud server will compute $\phi_{F,p_j} = H_0\left(H(F), k_{p_j}\right)$ and $\phi'_{F,p_j} = H_1\left(H(F), k_{p_j}\right)$ for each p_j satisfying $\mathbb{R}\left(p, p_j\right) = 1$ and $p \epsilon P_F$ will return to the user. Lastly, user computes the encryption $C_F = Enc_{CE}(k, F)$, where k random key which will be encrypted into ciphertext C_{k,p_j} with each key in $\left\{k_{F,p_j} = \phi_{F,p_j} \oplus H_2(F)\right\}$ using a symmetric encryption algorithm. And at end the user uploads $\left\{\phi'_{F,p_j}, C_F, C_{k,p_j}\right\}$ to end up the file uploading process.

4.4 File Retrieving

File retrieving procedure used similar to the construction method as discus above; suppose a user wants a file F to download. First step is to sends a request along with the file name to the public cloud. On receiving the request of data access and file name to public cloud, public cloud will first check whether user is eligible to download file F. If result failed means user is not a suitable person to download, then public cloud sends back an abort signal to user to indicate that download failure and can't access the respective data file. Otherwise, public cloud returns C_F a corresponding ciphertext for that data access file. On receiving the encrypted data from public cloud, the user uses the key k_F and stored locally to recover the original file F for which he/she had requested to public storage.

4.5 Auditing

Public auditing scheme in our system consists of four algorithms which are $(KeyGen, SigGen, GenProof, VerifyProof)$. Key generation algorithm that is run by user at time of setup as. $SigGen$ is which user used to generate verification of metadata along consisted of MAC signatures or other related information that will be used for the purpose of auditing in system. Cloud server runs $GenProof$ to generate a proof of data storage correctness. And the last one $VerifyProof$ is run by the TPA to audit the proof from cloud server which is a public storage in our case.

4.5.1 Setup

With executing $KeyGen$ the user initializes first the public and secret parameters of the system. Now pre-processes the data file F by using $SigGen$ which will generate the verification of metadata for the particular data content. Then user with the data file F and verification metadata stores to public storage.

4.5.2 Audit

The role of TPA is issues an audit message or challenge to public storage to make sure that public storage will retained data file F properly at the time of audit as a result. Public storage will derive a response message from a function of the stored data file F and its verification metadata by executing $GenProof$, as a result of response massage. The TPA then runs $VerifyProof$ to verify the response as auditing role.

5. IMPLEMENTATION

We implemented prototype of proposed authorized deduplication system with auditing, in which we used three model entities as separate java packages. *Cloud user* is data user which performs to carryout both upload/down file on public storage. A *Private Cloud* is used as private cloud which manages the private keys and handles the file token computations. A *Public storage* wills stores and checks duplicates files present in it.

We implement cryptographic operation of hashing and encryption/decryption with javax crypto packages [11] and for storage purpose to provide cloud computing environment cloudbus cloudsim packages [10] where used.

5.1 System Analysis

The security will be analyzed in authorization of duplicate check and confidentiality of data. For security of Duplicates check by considering antagonist for both internal and external, will try to break the system and by accessing cloud data or will illegal entrance to system.

5.1.1 Unforgeability of duplicate-check token

Assume a user with privilege p could forget new duplicate-check token $\emptyset'_{F,p'}$ for any p' that does not match p. If it's a valid

token, then it should be calculated as $\emptyset'_{F,p'} = H_1(H(F), k_{p'})$. Recall that $k_{p'}$ is a secret key kept by private cloud server and $H_1(H(F), k_{p'})$ is a valid message authentication code. Thus, without $k_{p'}$, the adversary cannot forge and output a new valid one for any file F . Any user with privilege p output a new duplicate-check token $\phi'_{F,p'}$, it also requires the knowledge of k_p. Otherwise, adversary could break security of message authentication code.

5.1.2 Indistinguishiability of duplicate-check token

Security of indistinguishability of token can be also proved based on assumption of underlying message authentication code is secure. Security of message for authentication code requires that adversary cannot distinguish if a code is generated from an unknown key. In deduplication system, all privilege keys are kept secret by private cloud server. Even if a user has privilege p, given a token $\emptyset'$, adversary cannot distinguish which privilege or file in the token because he/she does not have knowledge of privilege key sk_p.

5.1.3 Confidentiality of Data

Convergent key in construction is not deterministic in terms of the file. Depend on privilege secret key stored by private cloud server and unknown to the adversary. Thus, if adversary does not collude with the private cloud server, confidentiality of our second construction is semantically secure for both predictable and unpredictable file. Otherwise in case of, they collude then confidentiality of file will be reduced to convergent encryption because encryption key is deterministic.

Auditor success deals with proper handling the data content of the users with specific generated key.

5.2 EVALUATION

For testing of system, by compare three different parameter which are listed as 1) File size 2) Number of Files and 3) Deduplication ratio. By conduction experiment on Computer system with configuration as 32- bit OS, 3 GB ram and Intel core 2 Duo processor. Considering System for uploading of file which include process as 1) Tag file 2) Generate Token 3) Token access 4) Duplication Check 5) Uploading and 6) Auditing. By recoding time of start and end, crack up time for each process for different parameters, generating graphs for better understanding system

5.2.1 File Size:

To measure the file size effect, upload file which are unique to reduce the effect of deduplication. To study based on size of file the graph generated as shown in Fig. 2.

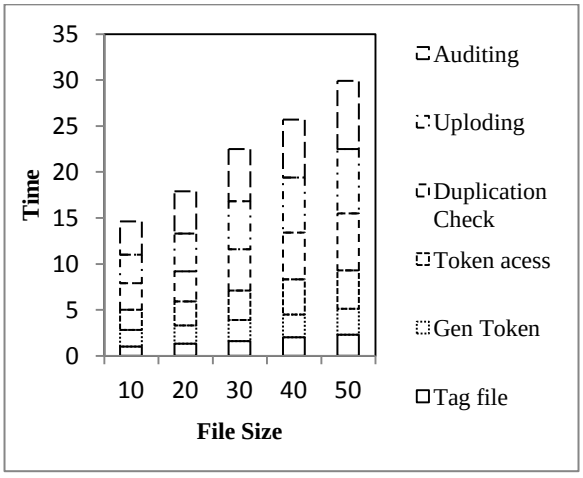

Figure 2. Crack up time for Different file size

5.2.2 Number of files stored

To measure the number of files stored, the effect on system with increased of files. For first, second processes the time doesn't increased in much faster but with token checking the time stamp increases in much faster rate.

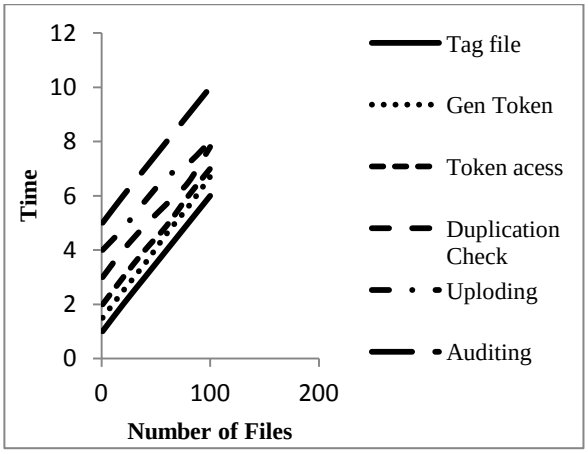

Figure 3. Crack up time for Different numbers of file

5.2.3 Deduplication Ratio

To measure the deduplication ration effect, with increase in deduplication percentage the uploading time reduces in much faster rate. That is the key advantage of our system. When the dedupliaction ratio is up to 100% the uploading time becomes zero. The total time spent on uploading a file with 100% dedupliaction ratio is only 41.26% as compare with the unique file.

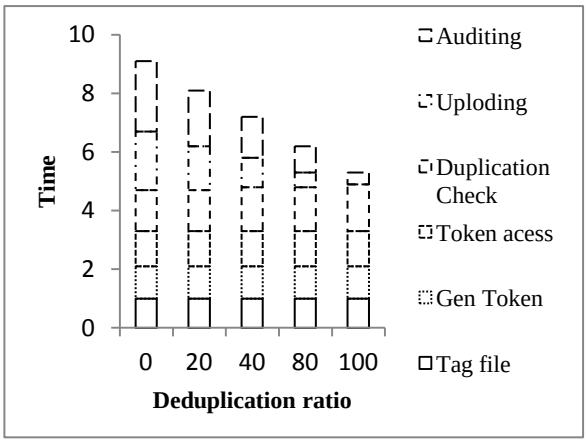

Figure 4. Crack up time for Different deduplication ratio

6. CONCLUSION

The popularity and widespread use of Cloud have brought great convenience for data sharing and collection. One critical challenge of cloud storage services is to manage the ever-increasing volume of data content stored. To make data managed scalable in cloud computing schemes, Data deduplication has been a well-known technique present. Data deduplication is a specialized data compression technique for eliminating duplicate copies of repeating data in storage. Although data deduplication contributes a lot of benefits, along with security and privacy concerns arise as user's sensitive data are susceptible to both insider and outsider attacks. In this paper, the idea of authorized data deduplication was proposed to protect data security by including differential privileges of users in the duplicate check. Compare from traditional deduplication systems, the differential privileges sets of users are further considered in duplicate check

besides itself the data. For support of stronger security the files are encrypted with differential privilege keys. The user is only allowed to perform the duplicate check for files marked with the corresponding privileges. The user can verify his/her presence of file after deduplication in cloud by auditing the data with the help of a third party auditor. The auditor audits and verifies the uploaded file on time. Therefore, the paper presents benefits to both the storage provider and user by deduplication technique and auditing technique respectively.

7. REFERENCES

[1] Yan Kit Li, Xiaofeng Chen, Patrick P. C. Lee, Wenjing Lou Jin Li, "A Hybrid Cloud Approach for Secure Authorized," *IEEE Transactions on Parallel and Distributed Systems*, vol. pp, pp. 1-12, 2014.

[2] S. Quinlan and S. Dorward., "Venti: a new approach to archival storage," *USENIX FAST*, Jan 2002.

[3] A. Adya, W. J. Bolosky, D. Simon, and M. Theimer. J. R. Douceur, "Reclaiming space from duplicate files in a serverless distributed," *ICDCS*, pp. 617-624, 2002.

[4] D. Harnik, B. Pinkas, and A. Shulman-Peleg. S. Halevi, "Proofs of ownership in remote storage systems.," *ACM Conference on Computer and Communications Security*, pp. 491-500, 2011.

[5] Sriram Keelveedhi, Thomas Ristenpart Mihir Bellare, "Message-locked encryption and secure deduplication," in *Springer Berlin Heidelberg,International Association for Cryptologic Research*, Advances in Cryptology – EUROCRYPT 2013, Athens, Greece, March 2013, pp. 296-312.

[6] S. Nurnberger, A. Sadeghi, and T. Schneider. S. Bugiel, "Twin clouds: An architecture for secure cloud computing.," *Workshop on Cryptography and Security in Clouds (WCSC 2011)*, 2011.

[7] Edward J. Coynek, Hal L. Feinsteink and Charles E. Youmank Ravi S. Sandhu, "Role-Based Access Control Models," *IEEE Computer*, vol. 29, pp. 38-47, Feb 1996.

[8] Sherman S.-M. Chow, Qian Wang, Kui Ren, and Wenjing Lou Cong Wang, "Privacy-Preserving Public Auditing for Secure Cloud Storage," *Computers, IEEE Transactions*, vol. 62, no. 2, pp. 362 - 375, Feb 2013.

[9] Chanathip Namprempre, Gregory Neven Mihir Bellare, "Security Proofs for Identity-Based Identification and Signature Schemes," *Journal of Cryptology,Springer-Verlag*, vol. 22, no. 1, pp. 1-61, January 2009.

[10] CLOUDS. [Online]. http://www.cloudbus.org/cloudsim/

[11] Crypto Packages Java. [Online]. http://docs.oracle.com/javase/7/docs/api/javax/crypto/package-summary.html