International Journal of Innovative Research in Science and Engineering
Vol. No.2, Issue 03, March 2016
www.ijirse.com
ISSN: 2454-9665

DETECT DATA DEDUPLICATION OVERHEADS FOR SAME DATA IN STORAGE

Swati Mali[1], Reshma Dhake[2], Roshani Torawane[3], Shweta Pawar[4], Guide: Prof.K.S.Kumawat[5]

[1,2,3,4]Student, Computer, Bvcoe&ri,

[5]Guide,Computer, Bvcoe&ri

ABSTRACT

De-duplication is a technique used to reduce the amount of storage needed by service providers. Now a day the most begining challenge is to perform secure de- duplication in data storage. Although encryption has been extensively accepted for secure de-duplication, a demanding issue of making encryption practical is to efficiently and reliably manage a massive number of keys. We first introduce a baseline approach in which each user holds an autonomous key for encrypting the keys and outsourcing them to the server. As a proof of concept, envelope the implementation framework of proposed authorized duplicate check scheme and conduct experiments using these prototype. In proposed system implement authorized duplicate check scheme sustain minimal overhead compared to normal operations. De-duplication is one of important data confining techniques for eliminating duplicate copies of repeating data. For that direction Authorized duplication check system is used. This paper

addresses problem of privacy preserving de-duplication in data storage and nominate a new de-duplication system supporting for Differential Authorization.In this project we are presenting the certified data deduplication to protect the data security by counting differential license of users in the duplicate check. Different new deduplication constructions presented for supporting authorized duplicate check. And also in this system hash key is generated with hash code.If two files are same and they have same hash key then second file which is same cannot shared to others.

.

Keywords: Deduplication, Distributed storage system , encryption, Key management ,Reliability.

I. INTRODUCTION

Deduplication techniques are highly employed to backup data and minimize network and storage overhead by detecting and removing redundancy among data, with the fast growth of digital data. Instead of keeping more than one data copies with the exact content, deduplication removes redundant data by keeping only one physical copy and referring other redundant data to that copy. Deduplication has appropriated much attention from both academia and industry because it can greatly make better storage utilization and save storage space, especially for the applications with high deduplication ratio such as expensive records storage systems. A number of deduplication systems have been pro-posed based on various deduplication bearing out plans such as client-side or server-side deduplications, file-level or block-level deduplications .

Data deduplication techniques become more fair and critical for the management of ever-increasing volumes of data in storage services which activate interest of enterprises and organizations to outsource data storage to third-party cloud suppliers , as clearly validated by many existing-life case studies . According to the particular structure of reported account of IDC, the volume of data in the world is demanded to reach 40 trillion gigabytes in 2020 .Today's commercial cloud storage services are Drop box, Google Drive and Mozy etc. They have been applying deduplication to deliver the network bandwidth and the data putting cost with client-side deduplication. There are two types of deduplication in advise of the size, first is file-level deduplication, which discovers redundancies between different files and removes these redundancies to cut capacity demands, and second one is block-level deduplication, which catch and removes redundancies between data blocks. The file can be splitted into two types as smaller fixed-size or variable-size blocks. The computations of block boundaries clarifies using fixed-size blocks or variable-size blocks. It adds better deduplication ability. Data reliability is a very captious conflict in a deduplication storage system because there is single copy for each file stored in the server. These files are typical by all the owners. If such a shared file was gone, a disproportionately large amount of data becomes remote because of the unavailability of all the files that share this file. In the additional, the challenge for data privacy also begins as more and users are being outsourced more responsive data to cloud. Encryption mechanisms have been used since out sourcing data into storage to conserve the confidentiality. Most commercial storage service provider is uncertain to apply encryption over the data because deduplication is impossible by it. The reason is that the conventional encryption working lack of different users to encrypt their data using their own keys, and containing public key encryption and symmetric key. However, at the cost of decreased error, these systems accomplished confidentiality of outsourced data.

II.OBJECTIVE

We are implementing the system using it stored multiple key for same data will be reduced. detecting duplicate copies of storage data before store it on storage.Validating whether data access is authorized when abnormal information access is detected, and Confusing the attacker with fake information.

III. PROBLEM STATEMENT

To reduce the deduplication we are going to implement technique using it we are reducing duplicate keys overhead as well as duplicate file data on server by using AES and Hashkey. Using hashing technique we are detecting duplicate files before upload.The simple idea behind de-duplication is to store duplicate data (either files or blocks) only once. Therefore, if a user wish to upload a file (block) which is already stored, the system provider will add the user to the owner list of that file (block).De-duplication has proved to achieve high space and cost savings and many system storage providers are presently adopting it..On the other hand, de-duplication introduces new security risks. So there is need of secure deduplication.

IV. LITERATURE SURVEY

R. D. Pietro in 2012 [1] Proof of Ownership scheme that has all features of the state of the art solution while provoking only a fraction of the overhead qualified by the competitor. In second, the security of the current mechanisms confide on information rather than assumptions. The quality of our current system is supported by extensive benchmarking.In year 2013[2] System formalize a new cryptographic primitive, Message Locked Encryption. Under the key encryption decryption are performed itself developed from message. MLE provides secure deduplication , a goal recently achieved by numerous storage suppliers.J. R. Douceur in 2002[3] The Farsite distributed file system gives availability by replicating each file onto more than one desktop computers. This repetation consumes significant storage space, it is important to recycle the space where possible.M. Bellare in 2013[4]In DupLESS, user convert data into code under message-based keys obtained from a key-server via an blind PRF protocol.It enables users to store encrypted data with an existing system, and it have the service behave deduplication on their behalf, and yet achieves strong confidentiality guarantees. A. Juels [5]A POR scheme enables a file or back-up service (prover) to produce a concise proof that a user (verifier) can retrieve a target file F, that is, that the

file retains and reliably transmits file data sufficient for the user to recover F in its entirety.J. Li, X. Chen [6] In Symmetric Encryption explain that notion of security and scheme for Symmetric encryption in focus security framework. They give several differ notion of security and analyses the concrete complexity of reduction among them. In this Convergent Encryption explain mechanism to reclaim space

from this incidental duplication to form it available for controlled file replication.

V. EXISTING SYSTEM

The dissecting kinds of data for each user stored in the database and the demand of long term continuous assurance of their data security, the problem of authenticating truth of data storage in the database becomes even more challenging. Database is not decent a third party data storehouse. The data stored in the database may be frequently updated by the users, on with insertion, deletion, modification, appending, reordering, etc. One critical challenge of today's storage services is the administration of the ever-increasing volume of data. According to the analysis report of IDC, the volume of data in the wild is conventional to grasp 40 trillion gigabytes in 2020. The baseline approach suffers two critic deployment issues. First, it is ineffective as it will generate an enormous number of keys with the increasing number of users. especially, each user must accessory an encrypted key with each block of outsourced encrypted data copies, so as coming restore the data copies. Although different users may share the same data copies, they must have their own set of parallel keys so that no other users can access their files. Second, the baseline approach is inaccurate, as it have need each user to dedicatedly protect his own master key. If the master key is unintentionally lost, then the user data cannot be reclaimed; if it is negotiate by attackers, then the user data will be leaked.

VI. PROPOSED SYSTEM

When the user wish to upload & download the file from database storage at that time first user request to the web server for uploading file miserly one of only authorized user can upload the file to web server for that purpose it use the acceptance of ownership algorithm. User to maintain their ownership of data copies to the storage server. When file is uploaded it partitions into blocks i.e block size is 4KB by default. According to file size the block occurs. Each block consist of their own cipher text , token for the unique identification and private key. The data storage server cover all the uploaded files and DB profiler store all the metadata of the file.

Case 1: When file F1 & F2 are special the all the data will be store in the database in different blocks.

Case 2: If the file F1 =F2 it stores only one file in the database divert duplication of the data.

Case 3: If F1€F2 then it compare the blocks with data storage and only various blocks of both file will be store in the data.

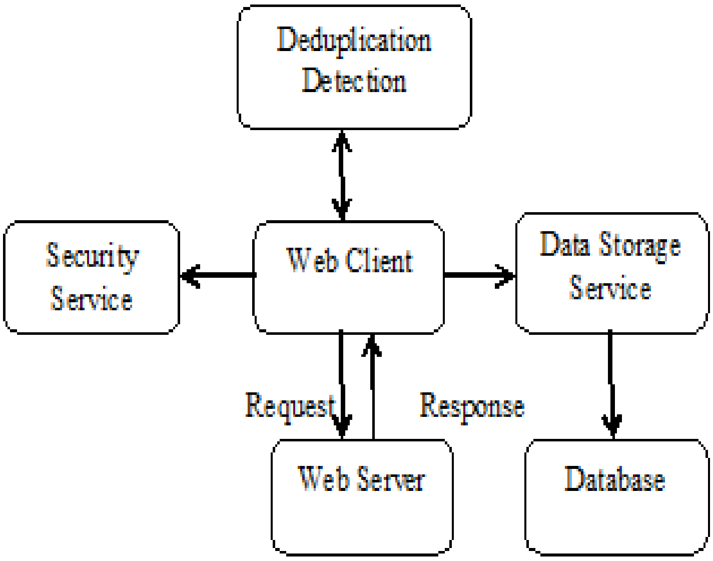


Fig-1:System Architecture

VII. ALGORITHM

7.1 The AES Encryption Flow

Data is a 128-bit block to be encrypted. The round keys are stored in Round Key Encrypt

```
Tmp = Add Round Key (Data, Round Key Encrypt [0])
For round = 1-9 or 1-11 or 1-13:
Tmp = ShiftRows (Tmp)
Tmp = SubBytes (Tmp)
Tmp = MixColumns (Tmp)
Tmp = AddRoundKey (Tmp, Round Key Encrypt [round])
end loop
Tmp = ShiftRows (Tmp)
Tmp = SubBytes (Tmp)
Tmp = Add RoundKey (Tmp, Round Key Encrypt [10 or 12 or 14])
Result = Tmp
```

7.2 The AES Decryption Flow (Using the Equivalent Inverse Cipher)

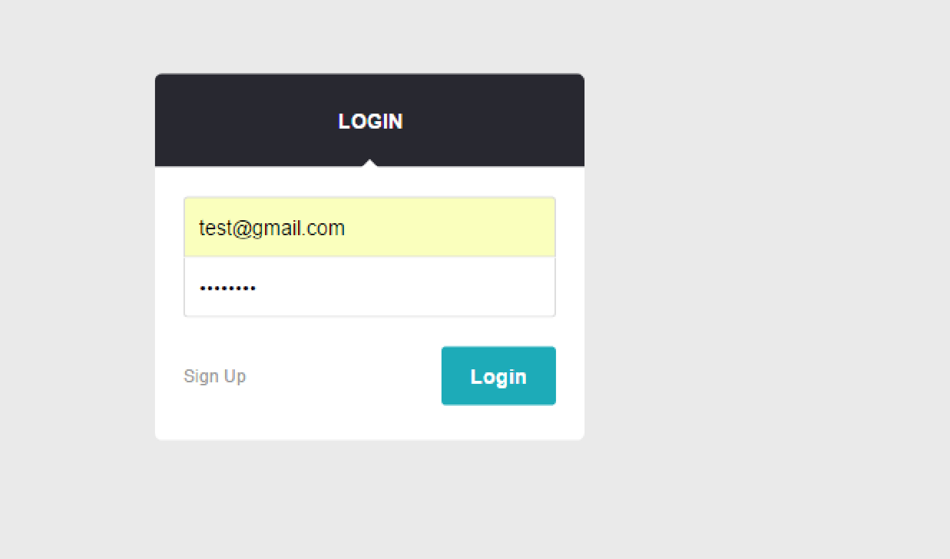
Data is a 128-bit block to be decrypt. The round keys are stored in Round Key Decrypt

```
Tmp = Add Round Key (Data, Round Key Decrypt [0])
```

```
For round = 1-9 or 1-11 or 1-13:
Tmp = InvShiftRows (Tmp)
Tmp = InvSubBytes (Tmp)
Tmp = InvMixColumns (Tmp)
Tmp = AddRoundKey (Tmp, Round Key Decrypt [round])
end loop
Tmp = InvShift Rows (Tmp)
Tmp = InvSubBytes (Tmp)
Tmp = AddRoundKey (Tmp, Round Key Decrypt [10 or 12 or 14])
Result = Tmp
```

VIII. IMPLEMENTED MODULE

Login Page

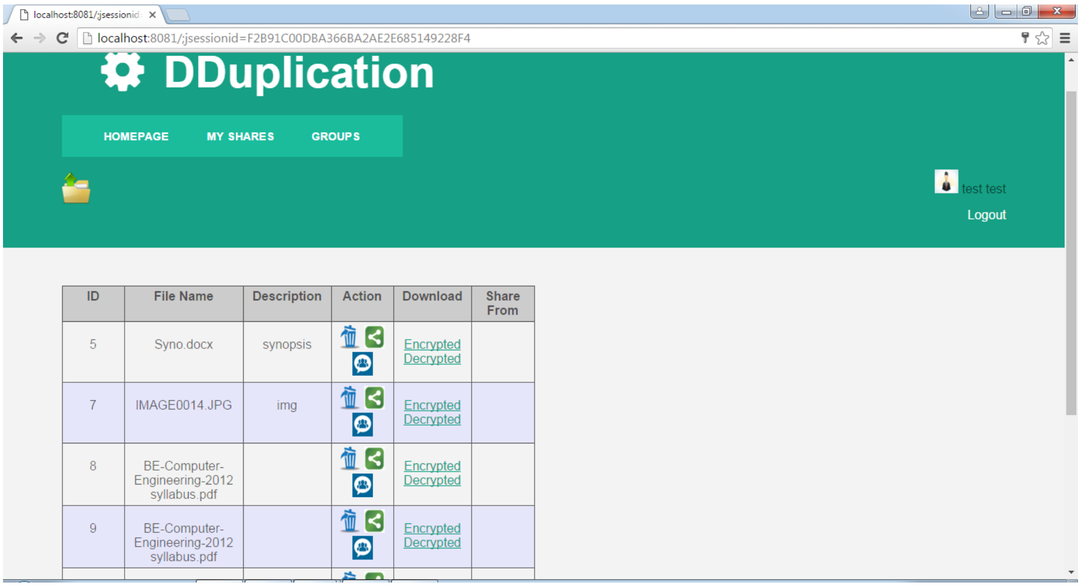


test@gmail.com

••••••••

Fig-1

In this system first user sign up on system. After successful sign in user get his user id and password. Then if he want to login then he can login on system .



localhost:8081/;jsessionid=F2B91C00DBA366BA2AE2E685149228F4

DDuplication

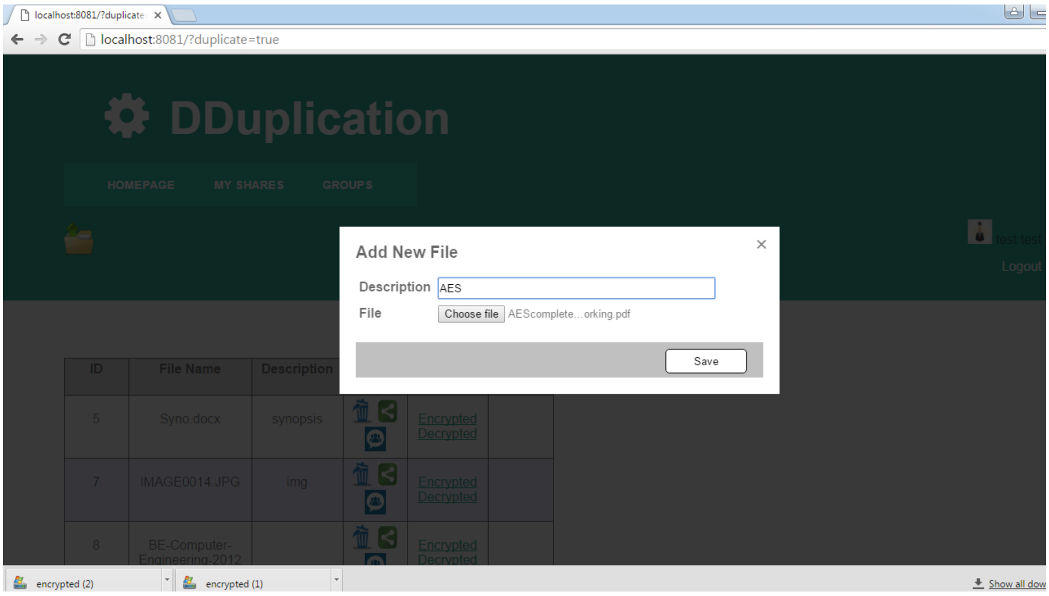
HOMEPAGE MY SHARES GROUPS

test test
Logout

ID	File Name	Description	Action	Download	Share From
5	Syno.docx	synopsis		Encrypted Decrypted	
7	IMAGE0014.JPG	img		Encrypted Decrypted	
8	BE-Computer-Engineering-2012 syllabus.pdf			Encrypted Decrypted	
9	BE-Computer-Engineering-2012 syllabus.pdf			Encrypted Decrypted	

Fig-2

After successful login homepage open on system. Homepage contain three things homepage, my shares,groups. User add file using add option and then user can logout from system.



localhost:8081/?duplicate=true

Add New File

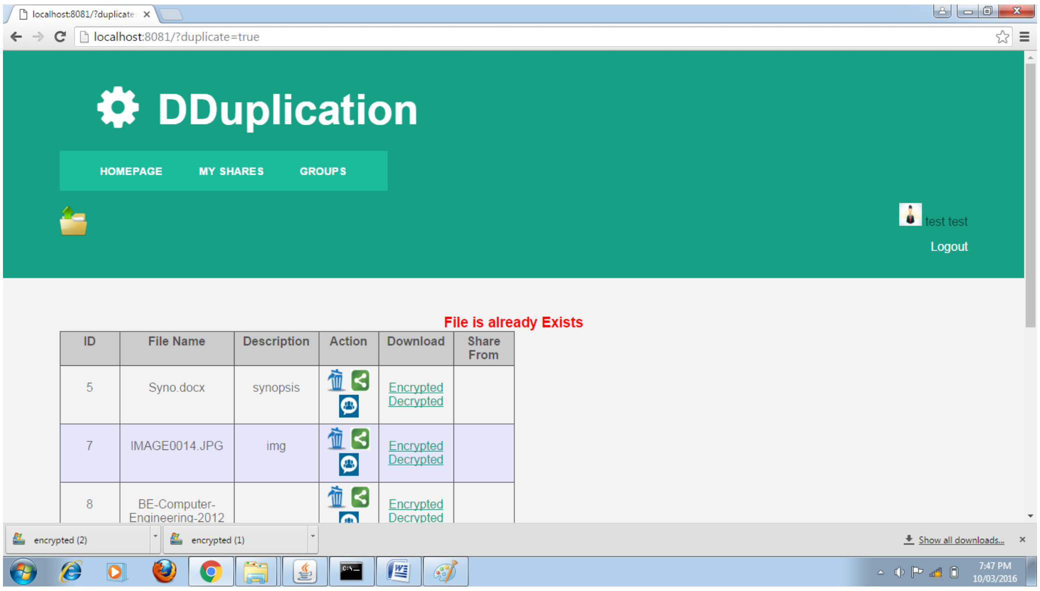
Description: AES

File: Choose file AEScomplete...orking.pdf

Save

Fig-3

User add file successfully in system. Then user can encrypt or decrypt this file before sharing to other user.

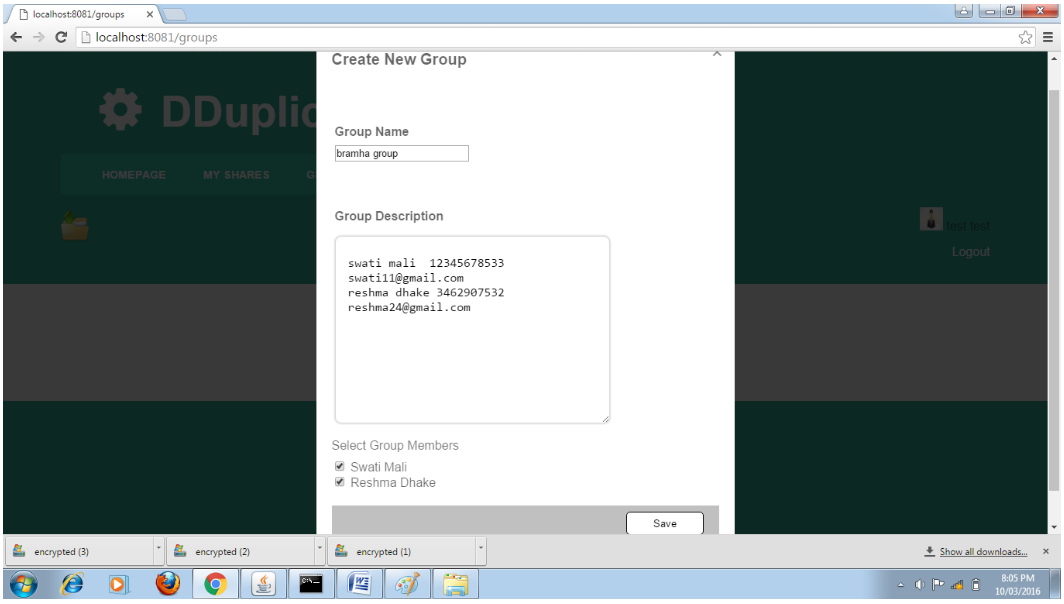


File is already Exists

ID	File Name	Description	Action	Download	Share From
5	Syno.docx	synopsis		Encrypted Decrypted	
7	IMAGE0014.JPG	img		Encrypted Decrypted	
8	BE-Computer-Engineering-2012			Encrypted Decrypted	

Fig-4

When file was add in system and this file is already present in system then system show file is already exists. Here deduplication is detected in system.



Create New Group

Group Name: bramha group

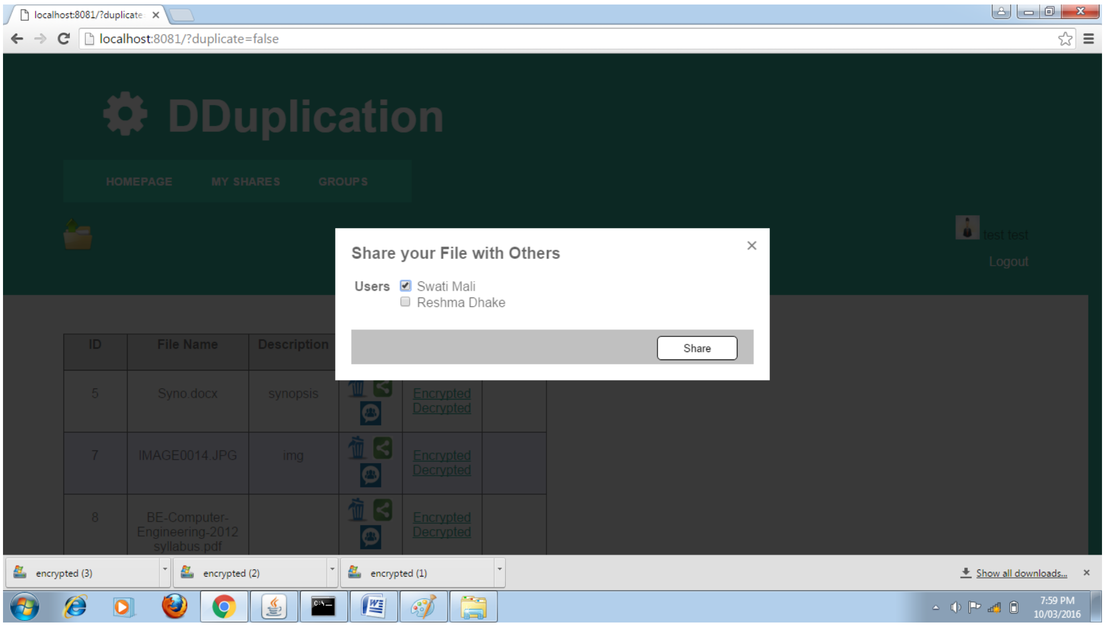
Group Description:

```
swati mali  12345678533
swati11@gmail.com
reshma dhake 3462907532
reshma24@gmail.com
```

Select Group Members

☑ Swati Mali
☑ Reshma Dhake

Fig-5



localhost:8081/?duplicate=false

Share your File with Others

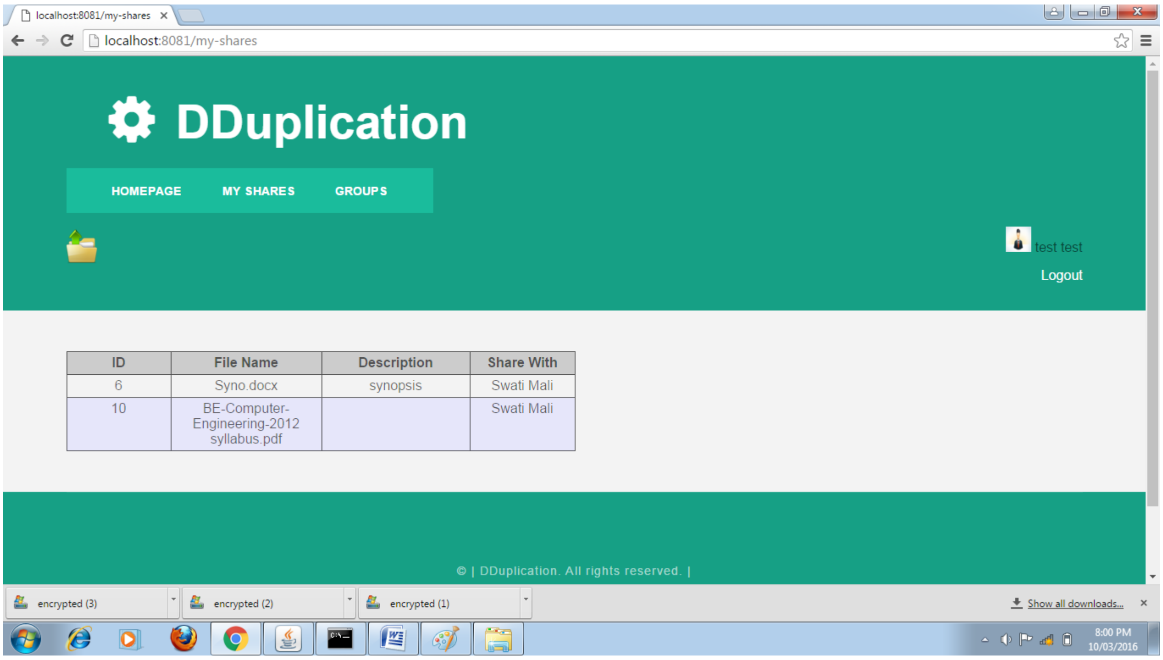
Users ☑ Swati Mali
☐ Reshma Dhake

Share

ID	File Name	Description		
5	Syno.docx	synopsis		Encrypted Decrypted
7	IMAGE0014.JPG	img		Encrypted Decrypted
8	BE-Computer-Engineering-2012 syllabus.pdf			Encrypted Decrypted

Fig-6

User creates groups in system. It is beneficial for user because when he want share a file at same time to other group members.In this module user select other user for sharing the file.



localhost:8081/my-shares

ID	File Name	Description	Share With
6	Syno.docx	synopsis	Swati Mali
10	BE-Computer-Engineering-2012 syllabus.pdf		Swati Mali

© | DDuplication. All rights reserved. |

Fig-7

User successfully share file to group and show names of other user which get share file.

8.1 Result analysis

User	Detect deduplication in %
User1	10
User2	20
User 3	30
User 4	40

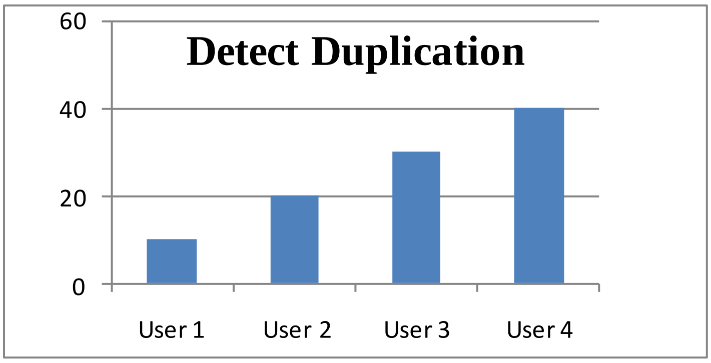


Fig - Result analysis

IX. CONCLUSION

We have completed our Encryption Decryption work in working application. We have achieved encryption using AES and hash value generated using SHA256 MD5 algorithm. The Advanced Encryption Technique was implemented successfully using Java language.different data messages were encrypted using different keys and varying key sizes. The original data was properly recruited via decryption of the cipher text. The modifications brought about in the code was tested and proved to be accurately encrypting and decrypting the data messages with even higher security and immunity against the unauthorized users.Security analysis demonstrates that our schemes are secure in terms of business executive in the making security model.As an indication of idea, we tend to enforced a paradigm of our planned approved duplicate check theme and conduct tested experiments on our paradigm. We tend to showed that our approved duplicate check theme incurs least overhead compared to merging secret writing and network transfer.

REFERENCES

[1] R. D. Pietro and A. Sorniotti, ”Boosting efficiency and securityin proof of ownership for deduplication.” in ACM Symposiumon Information, Computer and Communications Security, H. Y. Youmand Y. Won, Eds. ACM, 2012, pp. 81-82.

[2] ”Message-locked encryption and secure deduplication,” in EUROCRYPT,2013, pp. 296-312.615-1625.

[3] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer, ”Reclaiming space from duplicatefiles in a serverless distributed file system.” in ICDCS, 2002, pp. 617-624.

[4] M. Bellare, S. Keelveedhi, and T. Ristenpart, ”Dupless: Serveraided encryption for deduplicated storage,” in USENIX Security Symposium, 2013.

[5] A. Juels and B. S. Kaliski, Jr., ”Pors: proofs of retrievability for large files,” in Proceedings of the 14th ACM conference on Computer and communications security, ser. CCS ’07. New York, NY, USA: ACM, 2007, pp. 584-597. [Online]. Available:http://doi.acm.org/10.1145/1315245.1315317

[6] J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou, ”Secure deduplication with efficient and reliable convergent key management,” in IEEE Transactions on Parallel and Distributed Systems, 2014, pp. vol.25(6), pp. 1615-1625