Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IMPACT FACTOR: 6.017

IJCSMC, Vol. 6, Issue. 6, June 2017, pg.80 – 88

Secure Data Deduplication in Cloud Storage Based on Dynamic Ownership Management Using Bayesian Method

Mr. Akshay Huded[1], Mr. Rohit Kaliwal[2]

[1]Department of Studies in Computer Network Engineering, VTU Belagavi, India

[2]Assistant Professor Department of Studies in Computer Network Engineering, VTU Belagavi, India

[1] akshayhuded777@gmail.com; [2] rohit.kaliwal@vtu.ac.in

Abstract— One of the most important and popular service is cloud data storage. Nowadays, it has become an essential factor to preserve the data storage for security and/or privacy. So encryption technique is preferred to secure the data storage, but it has some issues like data deduplication. Data deduplication scheme cannot work on encrypted data. In existing technologies and solutions of encrypted data deduplication suffers from lack of security and/or privacy. To overcome the above problem, we propose a scheme of data deduplication on encrypted data storage based on dynamic ownership management and re-encryption key using Bayesian method. It integrates cloud data deduplication with access control. We evaluate its performance based on the computer simulations.

Keywords— Data deduplication, Bayesian, RSA, Ownership management etc

I. Introduction

Cloud computing endeavors a novel way of data technology services by recalling different expedients like storage and allowing it to users depends upon their claims. CC feeds a big expedient pool by connecting network expedients together. It has some luscious properties like expandability, scalability, fault tolerance and pay as per the usage. The most popular cloud offer data storage services. Users of cloud upload their personal data and credential data to the datacenter of service provider and allow it to manage that data. The most important and popular cloud service is data. Towards sensitive data at CSP intrusions are not avoidable hence CSP cannot be fully trusted by cloud users. The loss of control over credential data leads to high information security risks, particularly data privacy compromises. Although cloud storage area is huge hence data duplication highly desolate network expedients, consumes more energy and intricate data governance.

As discussed above one of the major problems with data storage service are security or privacy and management of growing huge volume of data. Recently data deduplication is considered as point of attraction in providing scalability in cloud computing environment. Actually data deduplication is a technique that is specialized in data reduction by eliminating duplicate copies of repeatedly requested data for storage. It is considered as an important technique which helps in improving the storage capacity. In spite of storing multiple copies of data files with the same contents deduplication technique is applied to eliminate redundant data by making sure that only one copy of data is stored it can either block or file level. In file level deduplication redundant copies of same files are neglected [1]. Similarly with the case of block level deduplication to avoid redundant copies [2].

The problem with data storage service security is assured by the application of some of the strongest encryption algorithms like RSA, ECC. Data redundancy increases as the number of same file uploads by same or different users increases. Before sending for uploading the data is encrypted with the keys of respective user's. While storing or uploading, the data is decrypted by the server and verified for duplication. If the data decrypted is same which is already uploaded by same or different users then it performs deduplication by allocating random key for data duplicated to their respective users. Unnecessarily, duplication of consumes more network resources, consumes more energy and complicates management of data. Deduplication is the only solution which overcomes the above issues [3].

II. LITERATURE SURVEY

Method to manage the encrypted huge volume of data in cloud computing environment ownership challenge management. This flexibly update data and data sharing with the application of deduplication even when the user operating in offline. Encrypted data can be accessed only by their respective users who are having the symmetric keys by decrypting that data by their respective keys. With respect to experimental results this scheme is considered as more secure and efficient [4].

System that achieves block data confidentiality by enabling block-level deduplication simultaneously. Authors showed that this scheme is worth performing block-level deduplication compared to file level deduplication. This scheme gains in terms of storage spaces which are not affected by the overhead of huge data management, which is minimal [5]. Various deduplication techniques that removes the replicates of files from the cloud environment. During analysis, many limitations were found that reduces the efficiency and security of this process. To solve those limitations Attribute and Policy based dedupe system is proposed to enhance the security of this schemes and also provides security against violation, unauthorized accesses etc [6].

Cloud Computing environments like redundancy and three encryption methods for Deduplication of data have been discussed here. The paper provides detailed study of present scheme challenges and different approaches [7]. Most secure deduplication scheme that highly provides supports to client-side encryption without the requirement of any additional servers. Interestingly, the scheme is depends upon PAKE (password authenticated key exchange) protocol. This scheme provides better security guarantees than previous efforts [8].

Server side duplication scheme for encrypted information. Because Client-side information deduplication specifically assures multiple transfers of constant contents, it solely consumes more space in network storage for one upload [9]. Scheme to deduplicate encrypted data stored that is to be stored in cloud by considering ownership challenge and proxy re-encryption. It also provides access control facility to assure authentication. The simulation result shows that this scheme provides superior efficiency and effectiveness compared to other existing methodologies[10]. Server-side deduplication model for data encryption. This scheme includes secured ownership group distribution of keys along with access control service for authentication. In addition, the proposed scheme guarantees data integrity against inconsistency attack [11]. Method depending upon the time of data arrival to the cloud. This technique enhances the storage capacity and improves the performance by comparing it with the data storing before by using MD5 hash algorithm and stores only the unique data [12].

III. METHODOLOGY

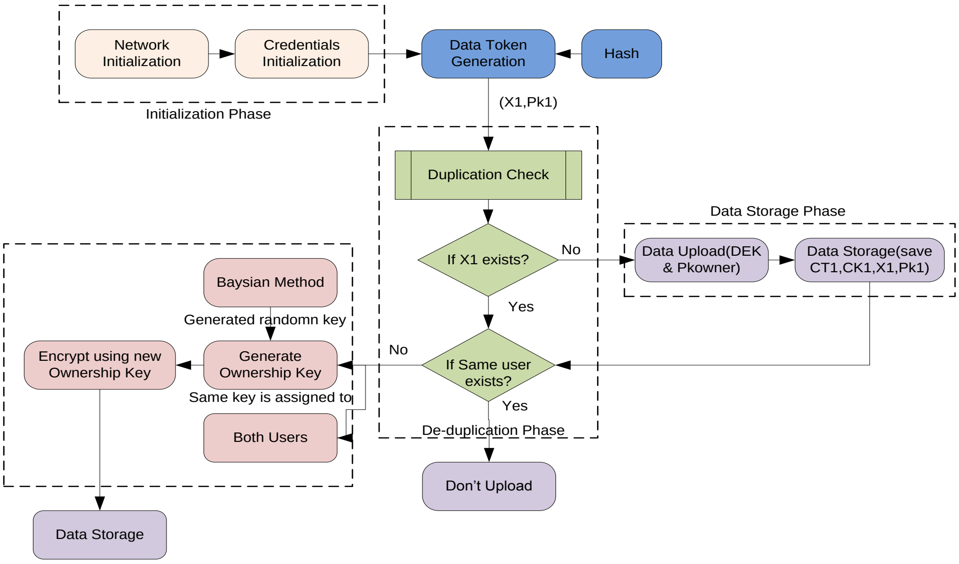


Fig.1 Architecture block diagram

The work is introduced here in the work that will generate the public keys using RSA method for user data encryption and XOR based encryption for the ownership encryption. This two level encryption processes the secure and fastest data downloading based on different cases. The steps followed here are same but to generate random keys for the different user of accessing the same file is chosen by the use of Bayesian method. The architecture block diagram of the proposed methodology is as shown in Fig.1.

A. Credential Initialization:

The process starts with assigning credentials to the number of users of the network. The credentials here are nothing but the private and public keys which are generated with use of an encryption technique called RSA.

B. Data token Generation:

When user want to upload data the data token of that user has to be generated. The token is of the form $x1 = H(H(M)) * P$, the $x1$ and Pki is sent to duplication check block. Hash function used here achieves authentication and access control features.

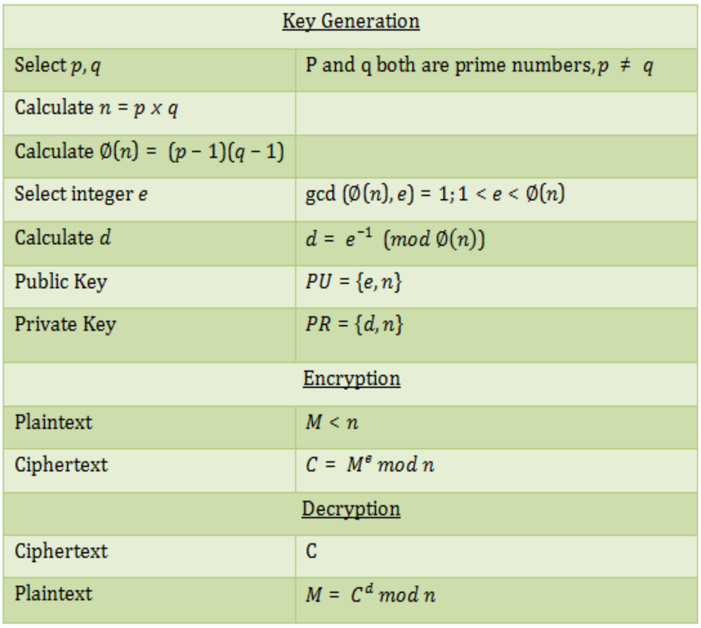
C. Duplication Check

It checks for the token, verifies whether x1 exists already, if no then only the user can upload data using data encryption key $(DEK1)$ to get cipher text $(CT1)$ and encrypt the $DEK1$ with Pk_{owner} to get cipher key $(CK1)$. In data storage save $(CT1)$,$(CK1)$, $x1$ and Pki. Else the deduplication is checked.

Once x1 exists, then we have to check for the user who has stored that file before, if the same user is only try to upload the same x1, then uploading is cancelled. Else for the encryption one random key is generated and selected based on the Bayesian method. Encryption is performed using the generated key and saved in data storage as new keys for both the users who have responded for the same data. Whenever request is made by the users the generated key is assigned for the further operations instead of assigning them different keys for encryption and decryption.

D. RSA Algorithm:

The design of an asymmetric public-private key in cryptosystem is accredited to Whitfield Diffie and Martin Hellman, who available the thought in 1976. Ron Rivest, Adi Shamir, and Leonard Adleman (R.S.A.) at MIT completed quite a lot of attempt supplementary the course of a year to produce with no return role with the aim of is hard to invert. Rivest and Shamir, as mainframe scientists, planned loads of probable function even as Adleman, as a mathematician, was responsible for finding their weaknesses. RSA algorithm is predicated on Diffie-Hellman algorithm for key transmission. It involves two phases: Key generation, encryption and decryption. In key generation phase key pair is to be generated for the two parties who want to communicate with each other. The key pairs are based on the prime values what we choose. Larger the prime value more will be the protection of keys generated for the users.



Key Generation	
Select p, q	P and q both are prime numbers, $p \neq q$
Calculate $n = p \times q$	
Calculate $\emptyset(n) = (p-1)(q-1)$	
Select integer e	gcd $(\emptyset(n), e) = 1; 1 < e < \emptyset(n)$
Calculate d	$d = e^{-1} \ (mod\ \emptyset(n))$
Public Key	$PU = \{e, n\}$
Private Key	$PR = \{d, n\}$
Encryption	
Plaintext	$M < n$
Ciphertext	$C = M^e \mod n$
Decryption	
Ciphertext	C
Plaintext	$M = C^d \mod n$

Fig.2 Overview of RSA algorithm

Encryption can be of two types here. One is encryption with private key at sender side and decryption with public key of sender at receiver side. Other one is encryption with public key of designation at sender side and decryption with the private key of destination at receiver side. Here public keys of communicating users are made public. The second one is considered as more secure compared to first one because if the encryption key is captured by any eavesdropper, he is not able to guess the decryption key to decrypt the message. Because, for that the appropriate algorithms are preferred and what are the prime numbers used if the prime values chosen are very large then it becomes difficult for him to crack it. Fig.2 depicts algorithm overview.

In the case of different users same file access; to make the ownership authentication we have used Bayesian method. In this method unique key sequence is generated for the ownership authentication. This key will be passed to the two users for future file decryption mode.

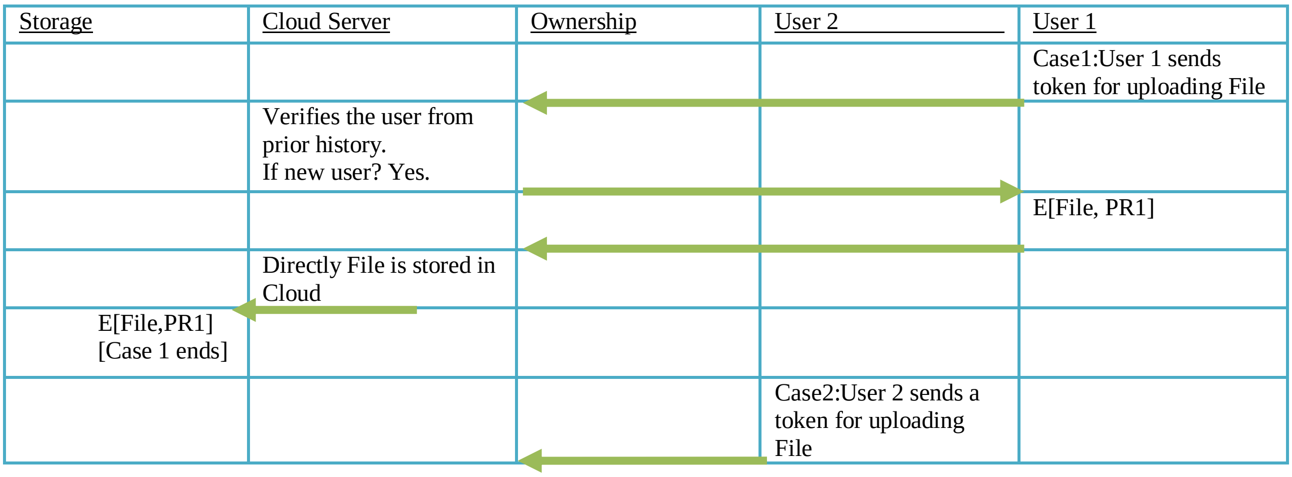
E. Bayesian Method

If we have the knowledge of real probabilities and the actual false positive as well as false negative probabilities, then we can modify the accurate answer for measurement errors. This is performed by relating the actual probability values to measured test probability. Bayes theorem will relate p (X|Y), the number of times that an event X happened for the total given the condition Y, and p(Y|X), the number of times the indicator Y happened for the given condition value that event X occurred. At the situation when; two users tries to save same file in the cloud to make a efficient utilization of storage system, we will generate a unique sequence using Bayesian. This key will be generated using the probabilistic flow as given in the Eq. (01) below

$$p\ (X|Y) = \frac{p(Y|X)\ p(X)}{p(Y|X)\ p(X) + p(Y|\sim X)\ p(\sim X)} = \frac{p\ (XY)}{p(XY) + p(\sim XY)} \qquad (01)$$

Where, probability conditions are checked for, key generated that is unique condition ration, probability of generating unique key in the overall system, probability of key generated as negative of the first case and finally the probability case for generating no unique key in the module. The calculations will provide a key to users based on the given set of three cases. We have generated a module for 100 unique key generations in the work and this can be extended as per the user requirements in the work with supportive modules. Figure 1 depicts the general working module of the proposed work.

IV. IMPLEMENTATION



Storage	Cloud Server	Ownership	User 2	User 1
				Case1:User 1 sends token for uploading File
	Verifies the user from prior history. If new user? Yes.			
				E[File, PR1]
	Directly File is stored in Cloud			
E[File,PR1] [Case 1 ends]				
			Case2:User 2 sends a token for uploading File	

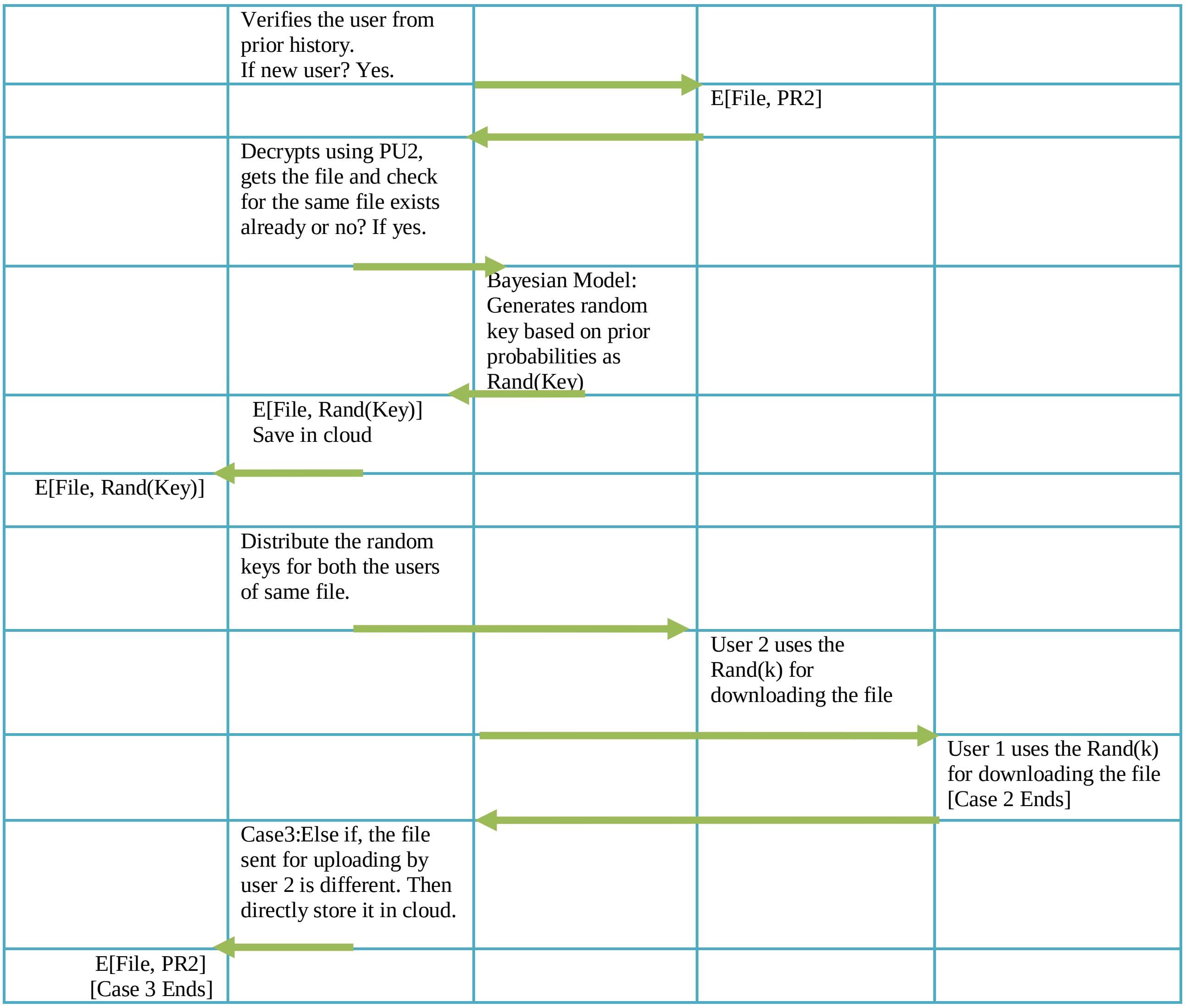

Verifies the user from prior history.
If new user? Yes.
E[File, PR2]
Decrypts using PU2, gets the file and check for the same file exists already or no? If yes.
Bayesian Model: Generates random key based on prior probabilities as Rand(Key)
E[File, Rand(Key)] Save in cloud
E[File, Rand(Key)]
Distribute the random keys for both the users of same file.
User 2 uses the Rand(k) for downloading the file
User 1 uses the Rand(k) for downloading the file [Case 2 Ends]
Case3:Else if, the file sent for uploading by user 2 is different. Then directly store it in cloud.
E[File, PR2] [Case 3 Ends]

Fig.3 Chart-flow of deduplication flow for different case

The different cases chart flow is presented here in Fig.3. The two user condition is the most important one in this work. Condition says that if same users are trying to save different files in the server then system should allow them to save the files in the cloud after single encryption of the files to be stored. If same users are willing to store same files in the cloud then the re-encryption will takes place by generated new public key for both users using the method called Bayesian. This module will create a public key using probabilistic equation that will not generate the same key for different conditions. Using this new key data packet is re-encrypted and stored in the cloud the key generated is passed to both users for the accessing purpose of the file in future. The simple flowchart with two user condition is represented in Figure 4 below and programmed in MATLAB.

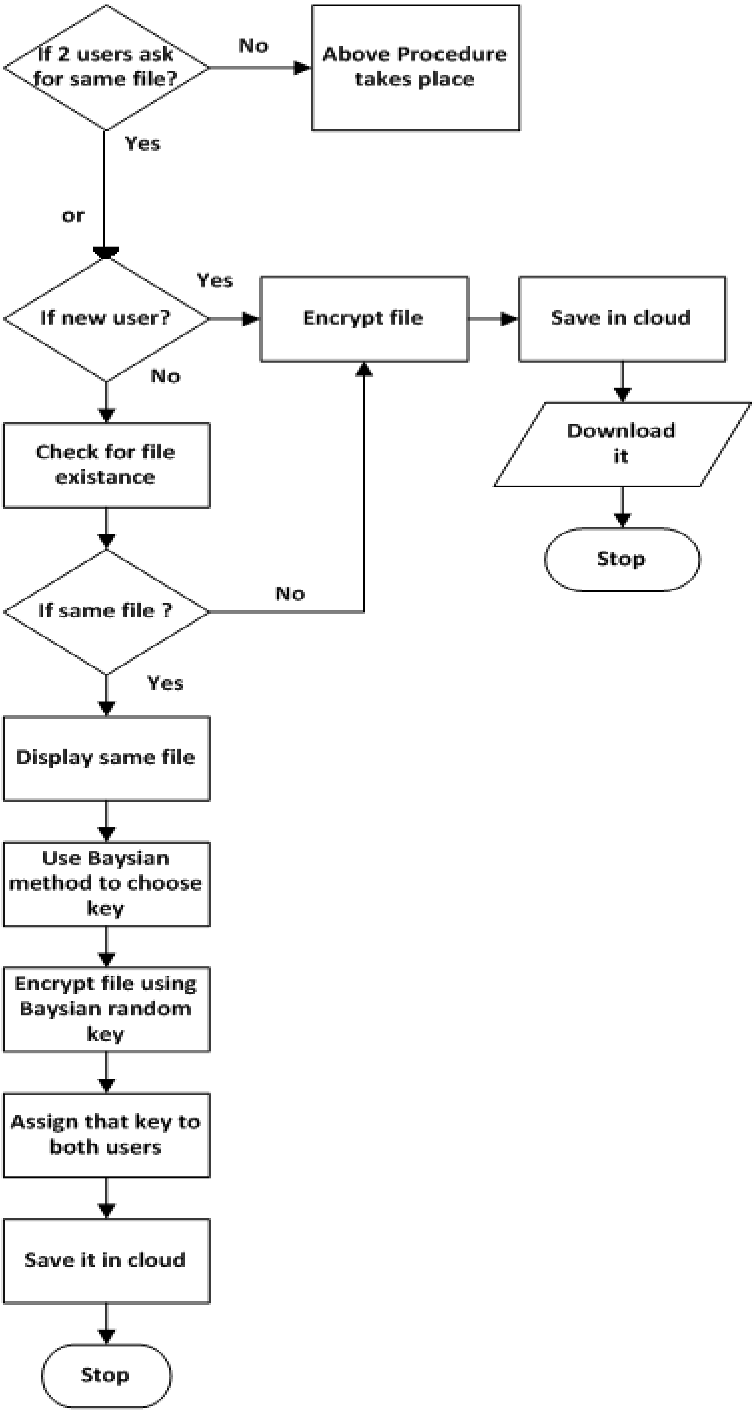

If 2 users ask for same file?
No
Above Procedure takes place
Yes
or
If new user?
Yes
Encrypt file
Save in cloud
No
Download it
Stop
Check for file existance
If same file ?
No
Yes
Display same file
Use Baysian method to choose key
Encrypt file using Baysian random key
Assign that key to both users
Save it in cloud
Stop

Fig.4 Flow chart for two users

V. EXPERIMENTAL RESULTS

The results of the complete work are depicted here. As we know the proposed work includes the different cases. The work begins by creating the network in work. The network is created by generating couple of users and server. Cases include the different conditions such as single user same file, single user different files, different users different files different users same file. In detailed explanation of the different cases are provided here in following section:

Case1. Same user and same file.

The same user and same file condition gives the basic flow of the work. If same users try to save same file in the server then the single copy is encrypted and stored in the server. The stored file will be encrypted using private key of the particular user. The key is used at the decryption mode when same user asks for the files stored the server using his private key.

Case2. Same user and different file.

When the situation arises at the source part that the same users are to be trying to save different files in the server that movement without any trouble the file will stored in the server after encryption

TABLE I

RESULTS FOR DIFFERENT CASES

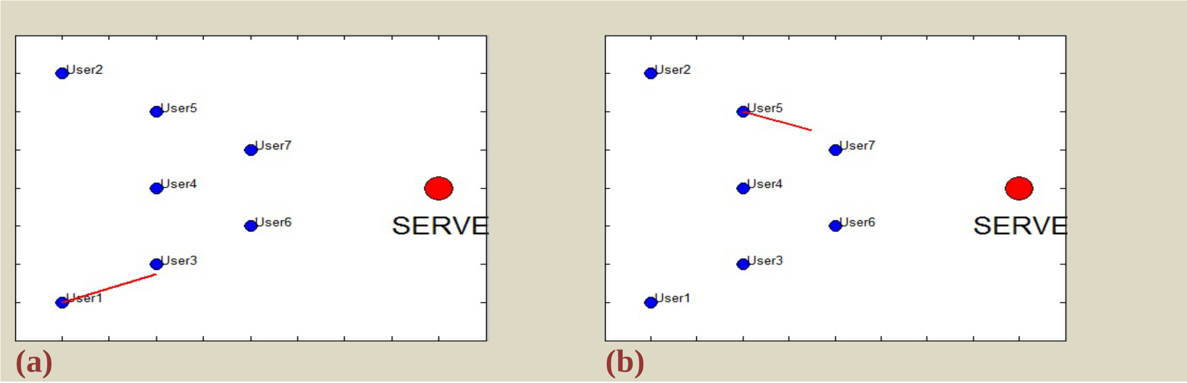
Decrypted File	Bayesian Key	File Re-Encrypted	File Encrypted	File Selected	2nd User	1st User
God is Great	-	-	RSA Encryption	God is Great	Owner 1	Owner 1
God is Great How r u	-	-	RSA Encryption	God is great How r u	Owner 1	Owner1
God is Great	Unique Key	Binary XOR Encryption	RSA Encryption	God is Great	Owner 2	Owner 1
God is Great How r u	-	-	RSA Encryption	God is great How r u	Owner 2	Owner1

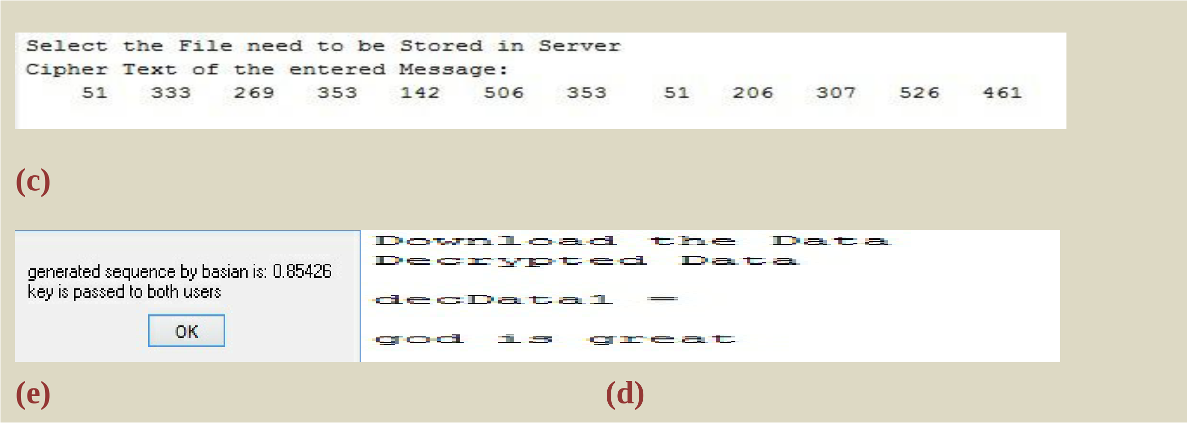
Case3.Different users same file.

This situation plays a very important role in the work. When two different users are willing to store the same file information need to be checked. If the files are same then as in the case1 single copy file has to be stored in the server but as users are different, server should make proper decision while passing the file to users. Hence second level encryption is made here by making use of Bayesian module. This Bayesian will generate sequence numbers those will never repeat and that unique sequence will act as a key to the module. The generated sequence is passed to the users and users will make use of this key at their respective decryption mode. The result of case 3 is presented in Fig.5.

Case4. Different users and different files.

When the situation arises at the source part that the different users are to be trying to save different files in the server that movement without any trouble the file will stored in the server after encryption. The detailed results of all the cases are given in the Table I.





```
Select the File need to be Stored in Server
Cipher Text of the entered Message:
    51    333    269    353    142    506    353     51    206    307    526    461
```

```
generated sequence by basian is: 0.85426
key is passed to both users
```

```
Download the Data
Decrypted Data

decData1 =

god is great
```

Fig .5 Results of the case 3, (a) user 1 transfers the file to server, (b) user 2 transfers the file to server, (c) encrypted file, (d) bayesian Key generated, (e) decryption performed.

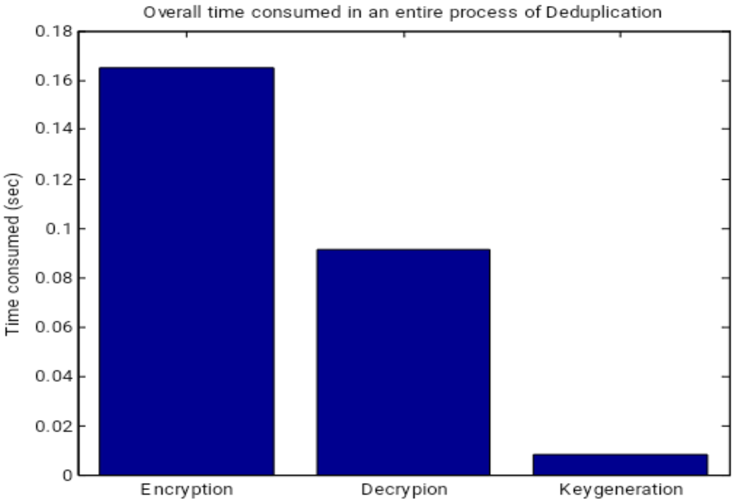


Fig. 6 Overall time consumed in an entire deduplication process

The performance of the system is analyzed by monitoring the time taken by the process for the intermediate steps such as, key generation, encryption and decryption. The proposed module is giving the better timing factors for the intermediate steps by providing efficient de-duplication of the data encryption, decryption and storing of the data in the server. Fig.6 is giving the timing factor details in the plot figure.

VI. CONCLUSION

Our scheme can flexibly support data update and sharing with de-duplication even when the data holders are offline. Encrypted data can be securely accessed because only authorized data holders can obtain the symmetric keys used for data decryption.

REFERENCES

[1] R.Karthikeyan and Mr. R.Velumani, " Data Deduplication with Security in Cloud Data Centres", International Journal of Innovative Research in Information Security (IJIRIS), Volume 5, Issue 2, 2015.

[2] Pasquale Puzio, Refik Molva, Melek O'nen, Sergio Loureiro, "Block-level De-duplication with Encrypted Data", Open Journal of Cloud Computing, Volume 1, Issue 1, 2014.

[3] Babaso D. Aldar and Vidyullata Devmane, " A Survey on Secure Deduplication of Data in Cloud Storage", International Journal of Innovation in Engineering and Technology, Volume 6, Issue 1, 2015.

[4] Pritee Patil, Nitin N. Pise, "Deduplication on Encrypted Big Data in Using HDFS Framework", International Journal of Innovative Research in Computer and Communication Engineering, Volume 4, Issue 10, 2016.

[5] Pasquale Puzio, Refik Molva, Sergio Loureiro, "ClouDedup: Secure Deduplication with Encrypted Data for Cloud Storage", IEEE, 2013.

[6] K. Keerthana, C. Suresh Gnanadhas, RT. Dinesh Kumar, "A Survey on Managing Cloud Storage using Secure De-duplication", Emerging Technologies in Networking and Security, Volume 7, Issue 9,2016..

[7] Francina Sophiya D and Swarnalatha P, "A Survey on Analysis of Efficient De-duplication in Cloud Computing Environment", International Journal of Computer Technology and Applications, Volume 9, Issue 26, 2016.

[8] Jian Liu, N. Asokan and N. Asokan, "Secure De-duplication of Encrypted Data without Additional Independent Servers", IEEE, 2016.

[9] Ashweta Magar, Trupti Jagtap, Pradnya Gaikwad, Rashmi Singh, " Avoiding Duplication of Encrypted Data Using Cloud", International Journal of Advanced Research in Computer and Communication Engineering, Volume 5, Issue 10, 2016.

[10] Zheng Yan, Wenxiu Ding, Xixun Yu, Haiqi Zhu, and Robert H. Deng, "De-duplication on Encrypted Big Data in Cloud", IEEE, 2016.

[11] Junbeom Hur, Dongyoung Koo, Youngjoo Shin, and Kyungtae Kang, "Secure Data Deduplication with Dynamic Ownership Management", IEEE, 2016.

[12] Zuhair S. Al-sagar, Mohammad S. Saleh, Aws Zuhair Sameen, "Optimizing the Cloud Storage by Data Deduplication: A Study", International Research Journal of Engineering and Technology, Volume 2, Issue 9, 2015.