ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 12, December 2017

Secure Dynamic Fragment and Replica Allocation of Data with Optimal Performance and Security in Cloud

Ashwini Yogesh Anikhindi[1], Prof M K Kodmelwar[2]

Designation Senior Coordinator, Department E& TC, College Symbiosis Lavale, Pune, India[1, 2]

ABSTRACT: High security systems are required to protect data within the cloud.Be that as it may, the utilized security technique should likewise consider the advancement of the information recovery time. We propose Secure Dynamic Fragment and Replica Allocation of data with optimal performance and security in cloud that collectively approaches the security and performance issues. In this methodology, we divide a file into fragments, and then replicate the fragmented data over the cloud nodes. Each of the nodes contains only a single fragment of a particular data file that ensures that even in case of a successful attack, no any meaningful information is disclose to the attacker. Furthermore, the nodes storing the fragments are separated with certain distance by means of graph T-coloring to bar from an attacker of guessing the locations of the fragments. Moreover, this methodology does not depend on the cryptanalysis techniques for the data security; thereby release the system of computationally costly methodologies. Moreover, we perform the downloading of the data can be done only at specific location and specific time and date. So that we can achieve maximum security as compare to other systems. We show that the eventuality to locate and settlement all of the nodes storing the fragments of a single file is extremely low. We also compare the performance of our methodology with ten other system. The comparison shows higher level of security with slight performance overhead was observed.

KEYWORDS: Centrality, Cloud Computing, Cloud Security, Fragmentation, Internet Protocol Vulnerability, Performance, Replication.

I. INTRODUCTION

Background

The cloud computing paradigm has remodeled the use and management of information technology infrastructures. Cloud computing is distinguished by self-service at the cards, ubiquitous network access, resource accumulation, elasticity, and uniform service. The above features of cloud computing make it a flashy candidate for companies, organizations and individual users for adoption. However, the benefits of low cost management, insignificant (from the point of view of the user), and greater flexibility have an increased security problem. Security is one of the most critical amongst those that bans the widespread adoption of cloud computing aspects. Cloud security issues may be due to the creation of basic technologies (escape virtual machine (VM), training session, etc.), offering cloud services (Structured Query Language injection, weak authentication schemes, etc). Recovery, vulnerability of Internet protocol, etc.). For a cloud it is safe, all participating entities must be safe. In any multi-drive system, the highest level of system security is the same as the low level security. Therefore, in a cloud, the security of goods depends not only on the security measure of an individual. Nearby entities can provide an opportunity for a malicious user to avoid user defenses.

Motivation

The data outsourced to a public cloud must be secured. Uncertified data access by other users and processes (whether accidental or intentional) must be prevented. Any weak entity can put the whole cloud at risk. In such a layout, the security mechanism must appreciably increase an attacker's effort to redeem a sensible amount of data even after a successful invasion of the cloud. Moreover, the probable amount of loss (as a result of data leakage) must also be minimized. A cloud must ensure throughput, reliability, and security. A key factor determining the throughput of a

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 12, December 2017

cloud that stores data is the data reclamation time. We design a new concept called Secure Dynamic Fragment and Replica Allocation of data with optimal performance and security in cloud that collectively approaches the security and performance issues. The proposed scheme ensures that even in the case of a successful attack, no concise information is revealed to the attacker. Not rely on traditional crypto analytics techniques for data security. The non-cryptographer nature of the proposed scheme makes it faster to perform the required operations (placement and retrieval) on the data. Shield a controlled replication of the file fragments, where each of the fragments is replicated only once for the purpose of improved security. A cloud storage security scheme collectively deals with the security and performance in terms of retrieval time.

Objective and Goal:

- Topropose scheme fragments and replicates the data file over cloud nodes.
- To increase both the security and performance.
- To achieve reliability, security, integrity of the data on the cloud.
- To ensure a restrain replication of the file fragments, where each of the fragments is replicated only once for the purpose of improved security.
- The proposed scheme protects that even in the case of a successful attack, no concise information is revealed to the attacker.

II. LITERATURE SURVEY

Juels et al., [2] presented a technique to make sure the integrity, novelty, and availability of data in a cloud.Data migration to the cloud is executed by the iris file system. An application gateway is designed and used in the organization to guarantee the integrity and novelty of the data using a Merkle tree. File blocks, MAC codes, and version numbers are kept at different levels of the plant. Additionally, the probable amount of loss in the event of a hardening date due to intrusion of access or other virtual machines may not decrease at.

G. Kappes, A. Hatzieleftheriou, and S. V. Anastasiadis, [3] presented problems related to virtualization and multi-tenancy in cloud storage using local combined storage access control and. The authorization architecture that combines the local access control Dike and the isolation of the host namespace are proposed.

D. Zissis and D. Lekkas, [5] presented the use of a trusted third party for providing security services in the cloud. Authors have used Public Key Infrastructure (PKI) to increase the level of trust in authentication, integrity (the drive), and the confidentiality of data and communication between the parties involved. Keys are generated and managed by certification authorities. At user level, the use of available test devices, such as smart cards, has been proposed to store keys.

D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, and A. Y. Zomaya, [6] proposed energy-efficient data replication in cloud computing datacenters.A central database (central DB) located on the WAN network provides all the data required by applications in the cloud. To speed up access and reduce latency, each datacenter houses a local database, called data center (Datacenter DB). It is used to duplicate the most used data elements of the central database. Each rack houses at least one server that can run a local rack of the database (DB Rack), used to replicate (duplicate) data center data centers.

Sabrina De Capitani di Vimercati1, Robert F. Erbacher2, [7] presented encryption and fragmentation for data confidentiality in the cloud which perform fragmentation of file.Fragmentation is to divide the attributes of a R report produce different vertical views (fragments) such that these views stored in non-secret external providers violate the requirements (directly or indirectly). Instinctively fragmentation protects sensitive association represented by a constraint attribute association c when c not all in the same fragment (publicly available) are, and fragments cannot be accessed by unauthorized users.

M. Tu, P. Li, Q. Ma, I-L. Yen, and F. B. Bastani, [10]presented a secure and optimal positioning of data objects in a distributed system. The encryption key is subdivided into actions n and is deployed to several sites within the network. Division of a n key action is carried out through the secret sharing threshold (k, n). The network is divided into groups. The number of duplicates and their location is determined through heuristic. A primary site is selected in each of the clusters that distribute replicas within the cluster.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 12, December 2017

Quanlu Zhang, Shenglong Li, Zhenhua Li, Yuanjian Xing, Zhi Yang, and Yafei Dai, [11] proposed CHARM: A Cost-efficient Multi-cloud Data Hosting Scheme with High Availability which integrates two key functions desired.The first is to choose different suitable clouds and a precise strategy for storing data redundancy at a minimal cost and guaranteed availability. The second is the precipitation of a transition process to redistribute data based on changes in access pattern patterns and price clouds.

Jian Liu, Kun Huang, Hong Rong, Huimei Wang, and Ming Xian, [12] proposed privacy-preserving public auditing for regenerating-code-based cloud storage.To resolve the problem of regeneration of failed authenticators in the absence of data owners, introduce a proxy that is hired to regenerate authenticators in the traditional public control system model. Also, testable public design is a verifiable automaton, which is generated by a pair of keys and can be regenerated with partial keys. Therefore, our scheme can completely free owners upload the data online. Additionally, the system assigns coefficients randomly to a pseudo-random coding function of data confidentiality gelatin.

Shristi Sharma, ShreyaJaiswal, Priyanka Sharma, Prof. Deepshikha Patel, Prof. Sweta Gupta, [13] are proposed an approach for file splitting and merging. File Splitter is a program that does not require installation and can be used to split files into multiple fragments, as well as to combine multiple fragments into a single file. File Splitter is software that is used to split the user definition file according to the specified size. It is very difficult to transfer a large file from one end to the other through any medium such as the Internet or small storage such as floppy, pen drive, CD, etc. This software helps solve this problem. Split file shares can take some temporary information to indicate the number of divided parts and the total number of parts, etc. This idea is used to divide large files into smaller parts in order to transfer, upload, etc. At the destination side, these file shares can be merged to form the original source file. The division process is mainly directed at the file transfer zone from one end to the other.

III. SOFTWARE REQUIREMENT SPECIFICATION

User Classes and Characteristics

To design products that satisfy their target users, a intense understanding is needed of their user characteristics and product properties in development related to unexpected problems that the user's faces every now and then while developing a project. The study will lead to an interaction model that provides an overview of the interaction between user characters and the classes. It discovers both positive and negative patterns in text documents as higher level features and employ them over low-level features (terms). In proposed work is designed to implement above software requirement. To implement this design following software requirements and hardware requirements are used.

Software Requirements

- Operating System - Windows XP/7
- Programming Language - Java/J2EE
- Software Version - JDK 1.7 or above
- Tools - Eclipse
- Front End - JSP
- Database - Mysql

Hardware Requirements

- Processor - Pentium IV/Intel I3 core
- Speed - 1.1 GHz
- RAM - 512 MB (min)
- Hard Disk - 20GB
- Keyboard - Standard Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - LED Monitor

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 12, December 2017

IV. COMPARISON BETWEEN EXISTING SYSTEM AND PROPOSED SYSTEM

In existing system data reliability, data availability, and response time are dealt with data replication strategies.Placing replicas data over a number of nodes expand the attack surface for that particular data.Existing system was not solving security and performance issues. We design a new concept called Secure Dynamic Fragment and Replica Allocation of data with optimal performance and security in cloud that collectively approaches the security and performance issues. The proposed scheme ensures that even in the case of a successful attack, no concise information is revealed to the attacker. Not rely on traditional cryptographic techniques for data security. The non-cryptographic nature of the proposed scheme makes it faster to perform the required operations (placement and retrieval) on the data. Ensure a controlled replication of the file fragments, where each of the fragments is replicated only once for the purpose of improved security. A cloud storage security scheme collectively deals with the security and performance in terms of retrieval time.

V. ALGORITHM FOR RELEVANT FEATURE DISCOVERY

- **File Fragmentation Algorithm:**

1. If file is to be split go to step 2 else merge the fragments of the file and go to step 8
2. Input source path, destination path
3. Size = size of source file
4. Fs = Fragment Size
5. NoF = number of fragments
6. Fs = Size/Nof
7. We get fragments with merge option
8. End

- **AES Encryption Algorithm**

Input: Key 128 bit and Data

Output: CipherText

Step1: Declare initVector=”RandomInitVector”
//This is 16 byte IV to generate the random key.

Step2: Create the objects
IvParameterSpec iv = newIvParameterSpec(initVector.getBytes("UTF-8"));
//UTF-8 is use to convert plaintext bytes into string
SecretKeySpecskeySpec = new SecretKeySpec(key.getBytes("UTF8"),"AES");
//SecretKeySpec class specifies a secret key

Step3: Create the objects
Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5PADDING");
// cipher class provides the functionality of a cryptographic cipher for encryption
//CBC-cipher block chaining i.e algorithm mode

Step4:
cipher.init(Cipher.ENCRYPT_MODE, skeySpec, iv);
byte[] encrypted = cipher.doFinal(value.getBytes());
//doFinal(byte[] input)-Encrypts data in a single-part operation, or finishes a multiple-part operation.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 12, December 2017

Step5:

Base64.encodeBase64String(encrypted));
//encodeBase64String(byte[] binaryData)-Encodes binary data using the base64 algorithm

Step6: End
// End the process.

- **Node Creation Algorithm:**

```
public void createNode(String path)
{
for(int i= 1;i<=16;i++)
{
if (!new File(path+"/"+i).exists())
      {
boolean status = new File(path+"/"+i).mkdirs();
System.out.println("Status  " +status);
System.out.println("node path:"+path+"/"+i);
      }
}
}
```

- **Fragment Placement Algorithm:**

Inputs and initializations:

$O = \{O_1, O_2, ..., O_N\}$

$o = \{sizeof(O_1), sizeof(O_2), ..., sizeof(O_N)\}$

$col = \{open\_color, close\_color\}$

$cen = \{cen_1, cen_2, ..., cen_M\}$

$col \leftarrow open\_color \forall$ i

$cen \leftarrow cen_i \forall$ i

Compute:

for each $O_k \in O$ **do**
  select $S^i \mid S^i \leftarrow$ indexof(max(cen_i))
  if $col_{S^i} = open\_color$ and $s_i >= o_k$ then
    $S^i \leftarrow O_k$
    $s_i \leftarrow s_i - o_k$
    $col_{S^i} \leftarrow close\_color$
    $S^{i'} \leftarrow distance(S^i, T)$ ▷ /*returns all nodes at distance T from S^i and stores in temporary set $S^{i'}$*/
    $col_{S^{i'}} \leftarrow close\_color$
  end if
end for

- **Replica Creation and Placement Algorithm:**

Inputs and initializations:
O = {O1,O2,….,ON}

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 12, December 2017

o = {sizeof(O1),sizeof(O2),…., sizeof(ON)}
col = {open color, close color}

```
for each O_k in O do
    select S^i that has max(R_k^i + W_k^i)
    if col_{S^i} = open_color and s_i >= o_k then
        S^i ← O_k
        s_i ← s_i − o_k
        col_{S^i} ← close_color
        S^{i'} ← distance(S^i, T)      ▷ /*returns all nodes at
        distance T from S^i and stores in temporary set S^{i'}*/
        col_{S^{i'}} ← close_color
    end if
end for
```

- **AES Decryption Algorithm**

Input: Key 128 bit and CipherText

Output: PlainText

Step1: Declare initVector=”RandomInitVector”
//This is 16 byte IV to generate the random key.

Step2: Create the objects
IvParameterSpec iv = newIvParameterSpec(initVector.getBytes("UTF-8"));
//UTF-8 is use to convert plaintext bytes into string

SecretKeySpecskeySpec = new SecretKeySpec(key.getBytes("UTF8"),"AES");
//SecretKeySpec class specifies a secret key

Step3: Create the objects

Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5PADDING");
// cipher class provides the functionality of a cryptographic cipher for decryption
//CBC-cipher block chaining i.e algorithm mode

Step4:cipher.init(Cipher.DECRYPT_MODE, skeySpec, iv);
byte[] encrypted = cipher.doFinal(value.getBytes());
//doFinal(byte[] input)- decrypts data in a single-part operation, or finishes a multiple part operation.

Step5: Base64.decodeBase64String(encrypted));
//encodeBase64String(byte[] binaryData)-Decodes cipher text using the base64 algorithm

Step6: End
// End the process.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 12, December 2017

VI. SYSTEM ARCHITECTURE

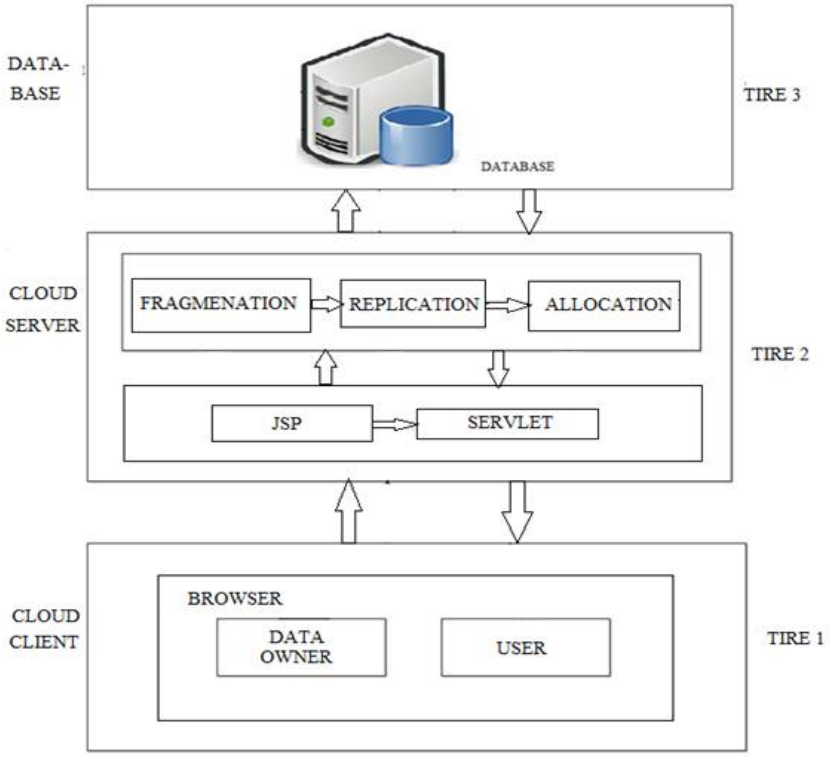


Figure 1: System Architecture

1) Cloud Client:

Cloud client should be Data owner or Data user.

- Data Owner:-
  Data owner is responsible for uploading file on cloud additionaly view files uploaded by him or others. Data owner has information about the placed fragment and its replicas with their node numbers in cloud.

- Data User:-
  Data user is the one who is responsible for downloading files or view files uploaded by others. To download file from cloud he has to be authenticated user otherwise he will be considered as attacker.

2) Cloud Server:

Fragmentation:-

This appeal is used for fragmenting the file for security purpose at sever side. This proposal runs the Fragmentation algorithm. It has file as input and produces the file fragments as output.

Replication:-

This appeal creates replicas (duplicate copy) of fragments. These replicas are useful when one of fragment is corrupted by attacker then to provide file for user admin replaces its replica at that place and combine all fragments and send file to authenticated user or data owner. To make replicas of file fragments this approach runs replication algorithm which takes input as fragments and produces its replicas as output.

Allocation:-

After the file is spitted and replicas are generated then we have to allocate that fragments at cloud server for storing data. While storing or allocating that fragments we have consider security issues. So we are using T-Coloring Graph concept for placing

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 12, December 2017

fragments at different nodules on cloud server. This approach runs Fragment allocation algorithm which takes input as fragments and produces the output as fragments allocated with node numbers.

3) Admin:
Admin is an accredited person who has rights to validate authorized data owner and user. He is also responsible for allocation of block and maintains information and authentication.

VII. EXPERIMENTAL SET UP AND RESULT TABLE

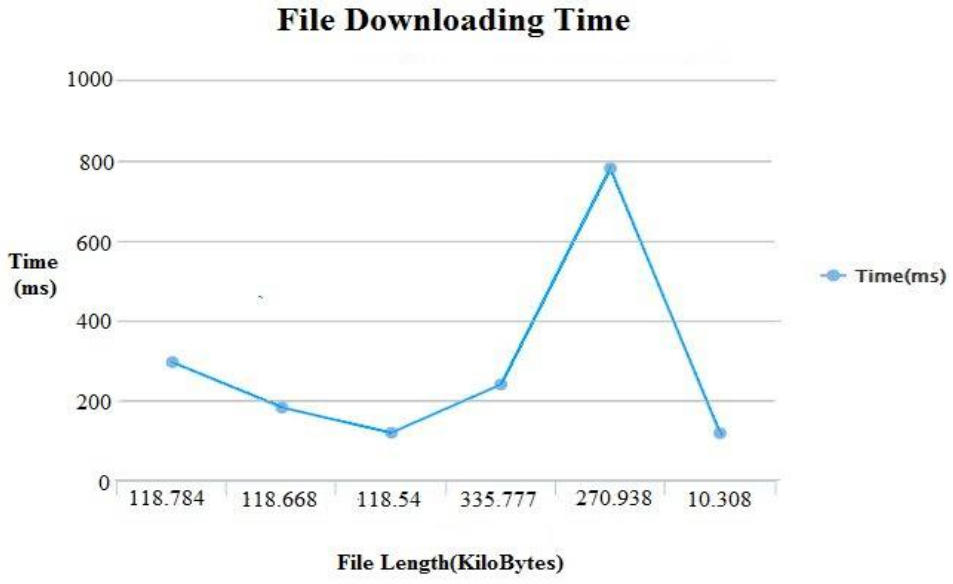
1. Result Table

File Length (Kilobytes)	Time(ms)
118.784	300
118.668	190
118.54	120
335.777	250
270.938	800
10.308	60

Table 1: File Downloading Time

Above table shows that how much time required for merging all fragments related to that file and downloads that particular file. We take file length i.e. size in kilobytes and time is in milliseconds (ms).

2. Result Graph



ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 12, December 2017

Figure 2: Result Graph of File Downloading Time

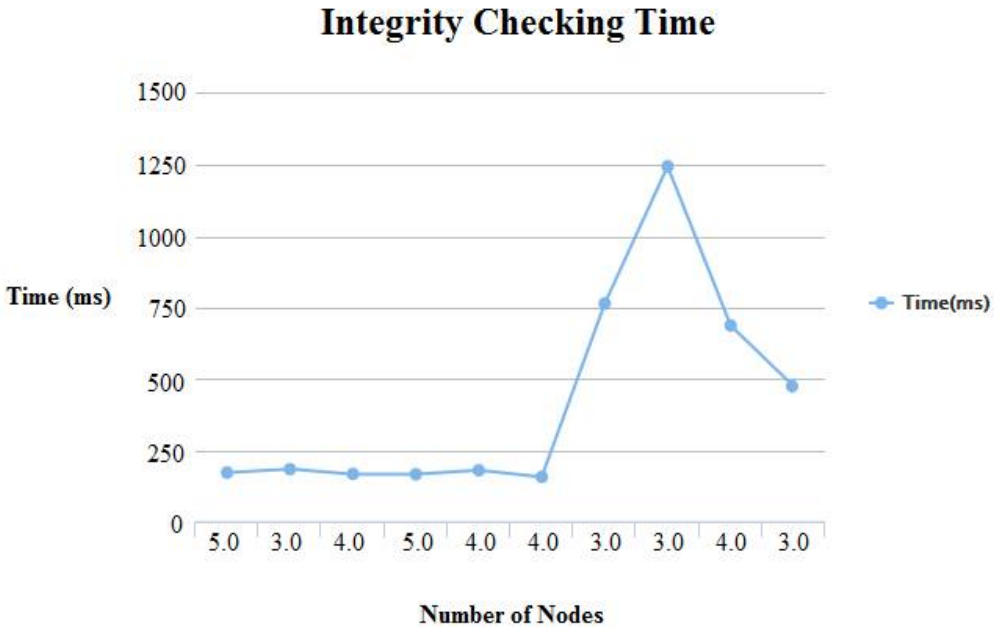


Figure 3: Result Graph of Integrity Checking Time

The above figure shows the File Downloading Time and Integrity Checking Time of proposed system. As can be seen in Figure 2 x-axis represent File Length in Kilobytes and y-axis represents time in milliseconds (ms). Figure 2 shows that how much time is required for merging all fragments and download the particular time .The above graph shows how much time system takes when user want to download file. In Figure 3 x-axis represents number of nodules of uploaded file and y-axis represents time in milliseconds (ms). Figure 3 shows that how much time is required for checking integrity of particular file by system.

VIII. CONCLUSION

We proposed the Secure Dynamic Fragment and Replica Allocation of data with optimal performance and security in cloud methodology, a cloud storage security scheme that collectively deals with the security and performance in terms of retrieval time. The data file was fragmented and the fragments are diffuse over multiple nodes. The nodes were separated by means of T-coloring. The fragmentation and dispersal ensured that no significant information was obtainable by an adversary in case of a successful attack. No node in the cloud, stored more than a isolated fragment of the same file. The performance of tour methodology was compared with full-scale replication techniques. The results of the simulations revealed that the simultaneous focus on the security and performance resulted in increased security level of data accompanied by a slight performance drop.

REFERENCES

[1] K. Bilal, M. Manzano, S. U. Khan, E. Calle, K. Li, and A. Zomaya, "On the characterization of the structural robustness of data center networks," IEEE Transactions on Cloud Computing,Vol. 1, No. 1, 2013, pp. 64-77.

[2] A. Juels and A. Opera, "New approaches to security and availability for cloud data," Communications of the ACM, Vol.56, No. 2, 2013, pp. 64-73.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 12, December 2017

[3] G. Kappes, A. Hatzieleftheriou, and S. V. Anastasiadis, "Dike: Virtualization-aware Access Control for Multitenant File Systems," University of Ioannina, Greece, Technical Report No. DCS2013-1, 2013.
[4] K. Bilal, S. U. Khan, L. Zhang, H. Li, K. Hayat, S. A. Madani, N. Min-Allah, L. Wang, D. Chen, M. Iqbal, C. Z. Xu, and A. Y. Zomaya, "Quantitative comparisons of the state of the art data center architectures," Concurrency and Computation: Practice and Experience, Vol. 25, No. 12, 2013, pp. 1771-1783.
[5] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," Future Generation Computer Systems, Vol. 28, No. 3,2012, pp. 583-592.
[6] D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, and A. Y. Zomaya, "Energy-efficient data replication in cloud computing datacenters," In IEEE Globecom Workshops, 2013, pp. 446-451.
[7] Sabrina De Capitani di Vimercati1, Robert F. Erbacher2, "Encryption and fragmentation for data confidentiality in the cloud".
[8] Y. Tang, P. P. Lee, J. C. S. Lui, and R. Perlman, "Secure overlay cloud storage with access control and assured deletion," IEEE Transactions on Dependable and Secure Computing, Vol. 9, No. 6, Nov. 2012, pp. 903-916.
[9] "Division and Replication of Data in Cloud for Optimal Performance and Security" azhar Ali, Student Member, IEEE, Kashif Bilal, Student Member, IEEE, Samee U. Khan.
[10] M. Tu, P. Li, Q. Ma, I-L. Yen, and F. B. Bastani, "On the optimal placement of secure data objects over Internet," In Proceedings of 19th IEEE International Parallel and Distributed Processing Symposium, pp. 14-14, 2005.
[11] Quanlu Zhang, Shenglong Li, Zhenhua Li, Yuanjian Xing, Zhi Yang, and Yafei Dai, "CHARM: A Cost-efficient Multi-cloud Data Hosting Scheme with High Availability". IEEE Transactions on Cloud Computing, Volume: 3March2015.
[12] Jian Liu, Kun Huang, Hong Rong, Huimei Wang, and Ming Xian, "Privacy-Preserving Public Auditing for Regenerating-Code-Based Cloud Storage". IEEE Transactions on Information Forensics and Security, Volume: 10, Issue: 7, July 2015.
[13] Shristi Sharma, ShreyaJaiswal, Priyanka Sharma, Prof. Deepshikha Patel, Prof. Sweta Gupta, "An Approach for File Splitting and Merging" Lecturer, Department of IT Technocrats Institute of Technology, Bhopal.