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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

# Secure Fragment and Replica Allocation of Data Using Geo-Social Attributes in Cloud

Nidhi Jain[1], Prof. Abhijit Janardan Patankar[2], Dr. Kshama V. Kulhalli[3], Dr.Kotrappa Sirbi[4]

M.E Student, Department of IT, Alard College of Engg, Murunji, Savitribai Phule Pune University, Pune, India[1].

H.O.D, Department of IT, D Y Patil College of Engg, Akurdi, Savitribai Phule Pune University, Pune, India[2].

Prof, Department of IT, D Y Patil C.E.T Kolhakur, Savitribai Phule Pune University, Kolhakur, India[3].

Principle, KLSs BCA/MCA RLS Institute, Belagavi, Associate professor, Department of CSE, KLEs Dr MSS College of Eng and Technology Belagavi, India[4].

**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 Fragment and Replica Allocation of information utilizing Geo-Social Attributes in cloud that all in all methodologies the security and execution issues.In this procedure, we isolate a document into sections, and afterward recreate the divided information over the cloud nodes.Each of the node contains just a solitary piece of a specific information record that guarantees that even in the event of an effective assault, no any important data is reveal to the assailant. Besides, the node putting away the sections are isolated with certain separation by methods for diagram T-coloring to banish from an aggressor of speculating the areas of the fragments.Moreover, this strategy does not rely upon the cryptanalysis strategies for the information security; in this way discharge the arrangement of computationally expensive methodologies.Moreover, we play out the downloading of the information should be possible just at particular area and particular time and date. With the goal that we can accomplish greatest security as contrast with different frameworks. We demonstrate that the projection to find and settlement the majority of the node putting away the pieces of a solitary document is to a great degree low.We additionally contrast the execution of our approach and ten other framework. The correlation demonstrates more elevated amount of security with slight execution 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.

DOI: 10.15680/IJIRCCE.2017. 0506203

**Motivation:**

Additionally, the plausible measure of misfortune (because of information spillage) should likewise be limited. A cloud must guarantee throughput, unwavering quality, and security. A key factor deciding the throughput of a cloud that stores information is the information recovery time. We plan another idea called Secure Fragment and Replica Allocation of information utilizing Geo-Social Attributes in cloud that on the whole methodologies the security and execution issues. The proposed scheme guarantees that even on account of a fruitful assault, no brief data is uncovered to the aggressor. Not depend on customary cryptoanalytics methods for information security. The non-cryptographer nature of the proposed conspire makes it speedier to play out the required operations (situation and recovery) on the information. Shield a controlled replication of the document sections, where each of the pieces is recreated once with the end goal of enhanced security. A cloud storage security scheme all things considered manages the security and execution as far as recovery 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 protect 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 is 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 cloud 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.

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-safeguarding open inspecting for recovering code-based cloud storage.To settle the issue of recovery of fizzled authenticators without information proprietors, present an intermediary that is procured to recover authenticators in the conventional open control framework demonstrate. Likewise, testable open plan is an unquestionable robot, which is created by a couple of keys and can be recovered with incomplete keys. Hence, our plan can totally free proprietors transfer the information on the web. Moreover, the framework allocates coefficients haphazardly to a pseudo-arbitrary coding capacity of information classification gelatin.

Shristi Sharma, ShreyaJaiswal, Priyanka Sharma, Prof. Deepshikha Patel, Prof. Sweta Gupta, [13] are proposed an approach for document part and consolidating. Record Splitter is a program that does not require establishment and can be utilized to part documents into different sections, and additionally to join numerous pieces into a solitary document. Document Splitter is programming that is utilized to part the client definition record as indicated by the predetermined size. It is exceptionally hard to exchange an extensive record from one end to the next through any medium, for example, the Internet or little stockpiling, for example, floppy, pen drive, CD, and so on. This product takes care of this issue. Split record offers can take some impermanent data to show the quantity of isolated parts and the aggregate number of parts, and so on. This thought is utilized to partition expansive records into littler parts to exchange, transfer, and so on. At the goal side, these record offers can be converted to shape the first source document. The division procedure is mostly coordinated at the document exchange zone from one end to the next.

## III. SOFTWARE REQUIREMENT SPECIFICATION

### User Classes and Characteristics

To design items that fulfill their objective clients, an exceptional comprehension is required of their client attributes and item properties being developed identified with startling issues that the client's faces from time to time while building up a venture. The examination will prompt a connection demonstrate that gives a review of the association between client characters and the classes. It finds both positive and negative examples in content records as larger amount elements and utilize them over low-level components (terms).In proposed work is intended to execute above programming necessity To implement this design following software requirements and hardware requirementsare 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 |

## IV. IMPLEMENTATION STATUS

**1) Fragmentation**

- In this first user registered and then login with the system
- After login successful user can be upload file on cloud
- While user uploading file on cloud first this module is worked
- We fragments input file
- To make fragments of file we have to first set the size for fragments
- Using the size file will gets original fragments

**2) Replication**

- After making a fragments we create copy of all fragments
- Those copy is called as replicas of fragments

**3) Fragments and Replication Placement**

- Once the file is split into fragments, our methodology selects the cloud nodes for fragment placement
- The selection is made by keeping an equal focus on both security and performance concerning the access time
- We choose the nodes that are most central to the cloud network to provide better access time
- For the aforesaid purpose, the Secure Dynamic Fragment and Replica Allocation of data with optimal performance and security in cloud methodology uses the concept of centrality to reduce access time
- For this purpose we are using this algorithm and T-coloring technique.

**4) Integrity Checking**

- Third party auditor (TPA) have an important responsibility i.e. to audit an integrity of file
- In our project TPA is responsible for auditing integrity of file
- File integrity is checking on the basis of generated hash value of each fragments
- If file is tempered on the node then system replace that tempered fragments by original fragments on node

## V. 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 Fragment and Replica Allocation of information utilizing Geo-Social Attributes 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.

## VI. 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.
```

**Step5:**

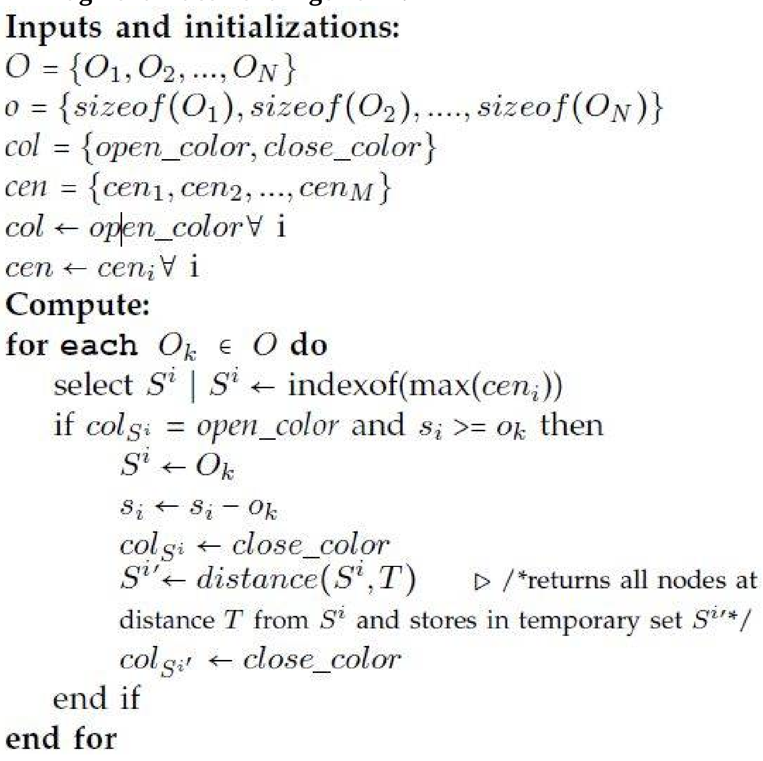
```
            Base64.encodeBase64String(encrypted));
            //encodeBase64String(byte[] binaryData)-Encodes binary data using the base64 algorithm
```

**Step6:** End
 // End the process.

- **Node Creation Algorithm:**

```
public void createNode(String path)
{
for(inti= 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);
        }
}
}
```

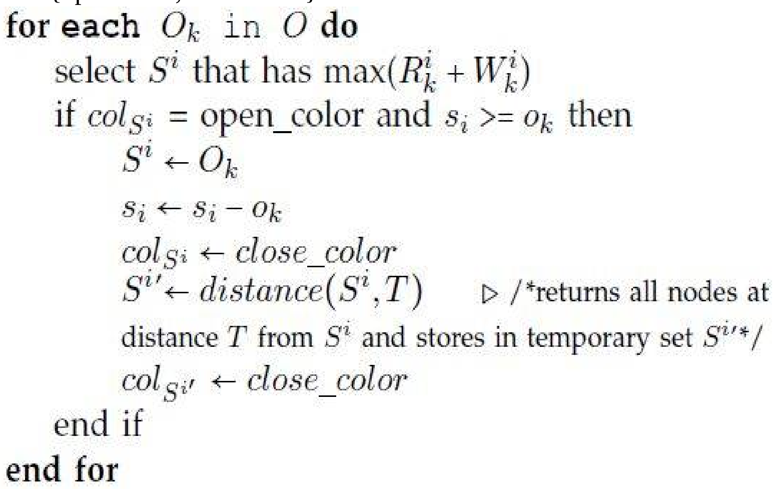
- **Fragment Placement Algorithm:**



```
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 $S^i$ | $S^i \leftarrow$ 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}
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
        $S^i \leftarrow O_k$
        $s_i \leftarrow s_i - o_k$
        $col_{S^i} \leftarrow close\_color$
        $S^{i'} \leftarrow distance(S^i, T)$    ▷ /*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
```

- **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.
```

## VII. SYSTEM ARCHITECTURE

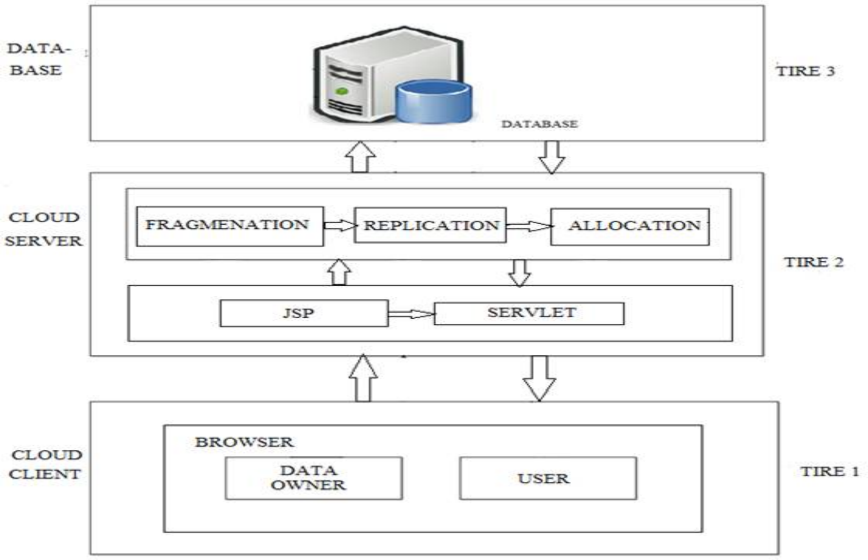


Figure 1: System Architecture

**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 authenticateduser 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:-

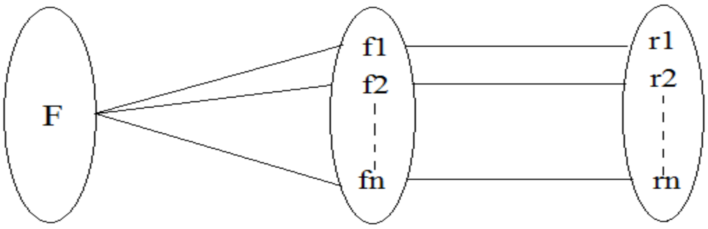
After the file is spitted and replicas are generated then we have to allocate that fragments at cloud server for storing data. While storing or allocating that fragments we have consider security issues. So we are using T-Coloring Graph concept for placing fragments at different nodules on cloud server.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

## VIII. MATHEMATICAL MODEL

### A] Mapping Diagram



Where,

F = File
f1…fn = Fragmentation of file
r1…rn = Replication of file

**B] Set Theory**

The following terms shows in detail working of project.

S={s, e, X, Y,*Φ*}

Where,

s = Start of the program.

1. Log in with webpage.
2. Load Files on cloud.

e = End of the program.

Retrieve the file from cloud storage system.

X = Input of the program.

Input should be File.

Y = Output of the program.

File will be first fragmented then it is replicated and the fragments are allocated using T-Coloring graph. Finally when we request for file downloading we get file as output.

X, Y ∈ U

Let U be the Set of System.

U= {Client, F, R, T}

Where Client, F, R, T are the component of the set.

Client=Data Owner, User

F=Fragmentation

R=Replication

T=T-coloring Graph

*Φ* = Failures and Success conditions.

**Failures:**

1. Huge database can conduct more time consumption to get the information.
2. Hardware failure.
3. Software failure.

**Success:**

1. Search the required information from available in Datasets.
2. User gets result very fast according to their needs.

## IX. 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

DOI: 10.15680/IJIRCCE.2017. 0506203

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

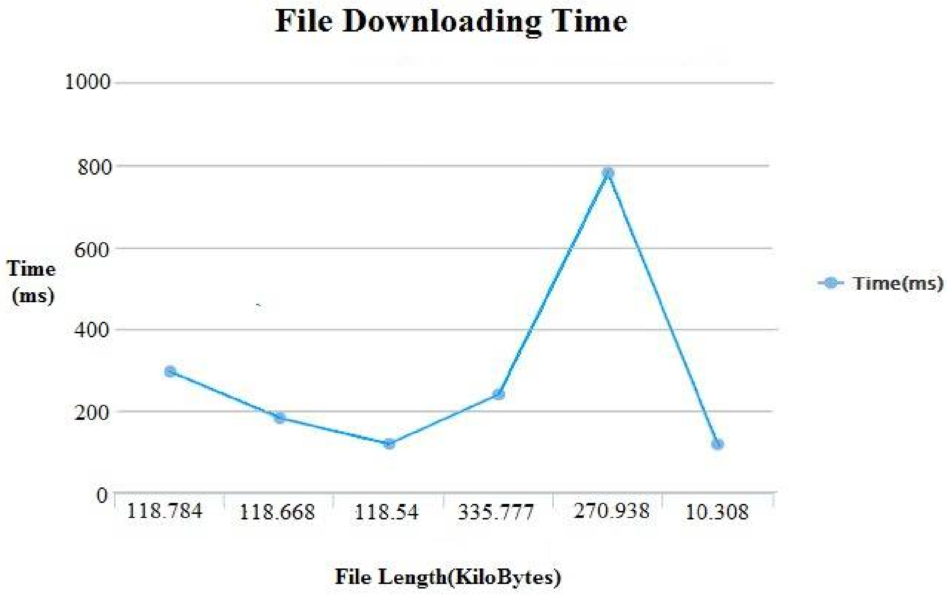


Figure 2: Result Graph of File Downloading Time

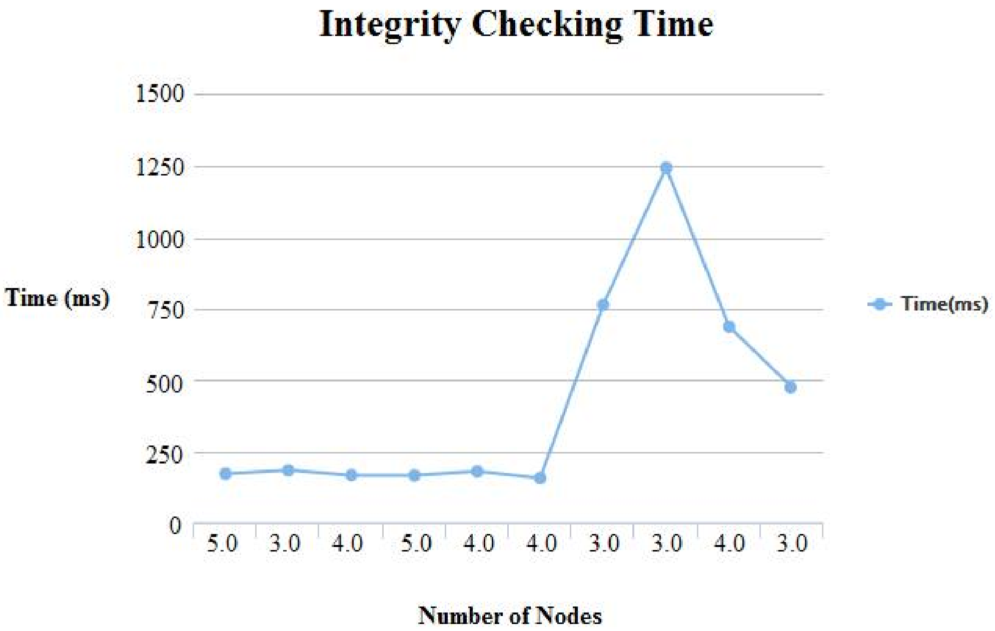


Figure 3: Result Graph of Integrity Checking Time

DOI: 10.15680/IJIRCCE.2017. 0506203

The abovefigure 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.

## X. CONCLUSION

We proposed the Secure Fragment and Replica Allocation of data using Geo-Social Attributes in cloudmethodology, 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.

[3] G. Kappes, A. Hatzieleftheriou, and S. V. Anastasiadis, “Dike: Virtualization-aware Access Control for Multitenant FileSystems,”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 Cloud 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.

DOI: 10.15680/IJIRCCE.2017. 0506203