ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2016

Data Manipulation Mechanism for Improved Security and Optimal Performance in cloud

Jyoti Bansode[1], Anjana Ghule[2]

PG Scholar, Dept. of Computer Science and Engineering, Government College of Engineering, Dr.Babasaheb Ambedkar Marathwada University, Aurangabad, India.[1]

Asst. Professor, Dept. of Computer Science and Engineering, Government College of Engineering, Dr.Babasaheb Ambedkar Marathwada University, Aurangabad, India.[2]

ABSTRACT: Distributed computing empowers client to store their information remotely and appreciate the on-interest incredible cloud application without the weight on nearby equipment and programming administration. The information trades off can happen on the grounds that assault by various client and hubs in the cloud. Accordingly, high security zones required to ensure data in the cloud. This paper gives high security and ideal execution approach for information stockpiling in cloud. In this system, when information proprietor needs to send document on cloud server, first record is part into pieces and after that encoding those sections. These encoded document sections over the cloud hubs. Each hub store one and only bit of particular information document to make ensure even if there should arise an occurrence of effective assault, no huge data is getting to the aggressor. Moreover, utilize the idea T-shading for putting away the parts in the hubs are detaching with certain separation. For keep up uprightness of information in cloud by utilizing Third gathering inspector and intermediary operator.

KEYWORDS: Cloud Security, Cloud storage, Fragmentation, Third Party Auditor, T-coloring.

I. INTRODUCTION

Cloud computing is a shared pool of configurable computing resources, on demand network access and provisioned by the service provider [2]. Cloud service provider will give services to cloud client. The significant issue in cloud data storage is to acquire rightness and integrity of information put away on the cloud. Cloud Service Provider (CSP) needs to give some type of mechanism through which client will get the confirmation that cloud information is secure. The significance of security is ample. The data housed on the cloud is regularly seen as significant to people with malicious intent. There is a great deal of individual data and conceivably secure information that individuals store on their PCs and this data is currently being exchanged to the cloud. This makes it basic for us to comprehend the efforts to establish safety that your cloud supplier has set up, and it is just as essential to take individual precautionary measures to secure the information.

Cloud storage auditing is utilized to confirm the integrity of the information stored openly cloud i.e. public cloud, which is one of the critical security procedures in cloud storage. So as to decrease the computational weight of the customer, Third party auditor (TPA) is acquainted with help the customer to intermittently check the integrity of the information in cloud. In this paper, we study the related work done on the Security and auditing of information stored in cloud storage in Section 2, the proposed work and implementation details in Section 3 where we see the system overview, Implementation details, mathematical model, algorithm and experimental setup and at last, we provide a result and conclusion in section 4 and Future scope in section 5.

II. RELATED WORK

Mazhar Al, Kashif Bilal, Samee U. Khan, BharadwajVeeravalli, Keqin Li, Albert Y. Zomaya[1] said Division and Replication of Data in the Cloud for Optimal Performance and Security (DROPS) that authoritatively sections client documents into pieces and reproduces them at vital areas inside of the cloud. The record division into sections is

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2016

performed in light of a given client criteria such that the individual parts does not comprise any important data. Each cloud node contains a particular piece to build the security for information. An effective assault on a single node must not uncover alternate pieces areas inside of the cloud. To enhance information recovery time, the nodes have chosen in light of the centrality measures that guarantee an enhanced access time. The nodes determination is performing in two stages. In the first stage, the nodes are selecting taking into account the pieces introductory arrangement on the centrality measures. In the second stage, the nodes are selecting for replication.

Jun Feng, Yu Chen, Wei-Shinn Ku, Zhou Su[3] propose a DOG (Data Division and Out-of request key-stream Generation), a substantial execution equipment usage arranged stream cipher for distributed data storage system. The D-DOG makes cipher block by dividing the plaintext information into various pieces and encrypting them, where the key-stream is created by abstracting bits from the data block in a pseudo arbitrary out of-request way. The DDOG keeps away from one of the shortcomings genuine in advanced stream cipher show up from the settled length in statement vector (IV). D-DOG creates the key-stream by separating n bits from the plaintext in a pseudorandom way. The key-stream length n is adaptable and maybe set by particular security prerequisites. The variable length key-stream makes brute force attack significantly more troublesome and the pseudo arbitrary piece abstracting makes decrypted information stream still unrecognizable unless the key-stream bits are embedded back to the first position.

Alessandro Mei, Luigi V. Mancini, and SushilJajodia[4] aims at designing a solution inview of countless correlatives to the decentralized algorithm all together that ensure the accessibility and the framework's functionalities versatility. The record framework administrations does not contain centralized server, just an arrangement of collaborating nodes to give data storage, all inclusive get to, and restore to remote clients in an adaptable and dynamically reconfigurable way. Only customers havetrusted while all servers are un-trusted; this change unequivocally influence to the security and accessibility models. In a fragmentation scheme, a file f is division into n sections, all pieces have marked and circulated to n remote servers, one piece per server. The client can recreate record f by getting to pieces selfassertively picked. The calculation works in the read-m-compose all setting. When all is said in done, m sections read are performed from the nearest servers among those that store the n record parts. This paper proposes a model to evaluate document confirmation put away in such a framework, where the record certification isthe likelihood for record has not been traded off under the presumption that the framework is the objective assault fruitful.

Parakh A, and Kak S [5] giving Implicit Storage Security to information in Online is more valuable in a cloud situation. Displayed certain capacity security building design for putting away information where security is scattered among numerous elements furthermore take a gander at some basic parceling routines. So information dividing plan is proposed for online data storage that includes the limited field polynomial root. This technique contains two dividing plan. Splitting data are saved money on cloud servers that are picked in an irregular way on system and these segments are recaptured with a specific end goal to remodel the expert duplicate of information. Information pieces are open to one who has learning of passwords and capacity areas of divided pieces.

Cong Wang, Sherman S.-M. Chow, Qian Wang, KuiRenandWenjing Lou [6]said the public auditing is an arrangement of data storage security in Cloud Computing and give a protection saving evaluating convention, i.e., this paper empowers an external auditor to review client's outsourced information in the cloud without taking in the information content. The public auditability is to permit TPA to confirm the accuracy of the cloud information on interest without recovering a duplicate of the entire information or acquainting extra online weight with the cloud clients. The Privacy-protecting is to guarantee that the TPA can't get clients' information content from the data gathered amid the inspecting process. In particular, this paper accomplishes group examining where numerous designated evaluating errands from various clients can be performed all the while by the TPA. In this paper ensure the security and legitimize the execution of our proposed plans through solid investigations and examinations with the best in class.

D.H. Patil Rakesh R. Bhavsar and AkshayS[7] said the data secrecy, confirmation and access control issues in distributed proposing so as to compute have been tended to a structure to build the cloud reliability and dependability. A cryptography algorithm Diffie-Hellman for secure correspondence, rather than key dispersion administration, is used in this work. Such a framework typically comprises of three modules: organization, verification and encryption modules. Every module has diverse, yet interconnected, capacities. The administration module is utilized for authentication of clients, and encryption module is utilized for information encryption. The confirmation acknowledgment is a two - way handle. Firstly, the framework requires the client to enter ordinary login and secret word, and after that it creates one-time watch word and sends it on the client versatile for verification. Once the one-time secret word is supplied, the framework confirms the client and awards access to the framework.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2016

Deshmukh P M, Gughane A S et al [10]. assure the security of stored data in cloud, presented a systemwhich utilizes distributed scheme. Proposed system consists of a master server and a set of slave servers. There is no direct communication link between clients and slave servers in the proposed model. Master server is responsible to process the clients requests and at slave server chunking operation is carried out to store copies of files in order to provide data backup for file recovery in future. Users can also perform effective and dynamic data operations. Clients file is stored in the form of tokens on main server and files were chucked on slave servers for file recovery. Thus proposed scheme achieved storage correctness insurance and data availability by using Token generation
algorithm with homomorphic token and merging algorithm were used.

III. PROPOSED WORK

1.1. SYSTEM OVERVIEW

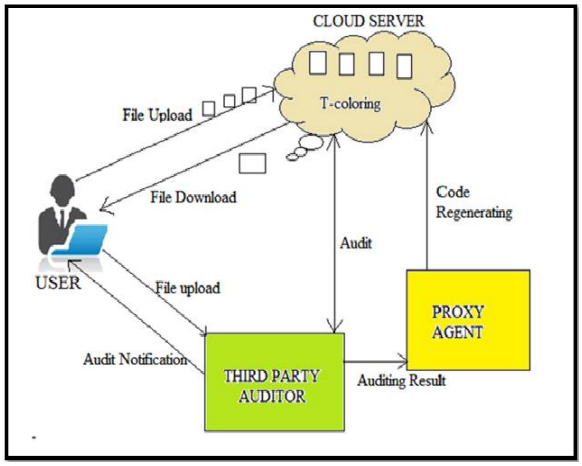


Figure1. System Architecture

Many Existing Systems were challenging to provide integrity checking against the service attacks and threads over cloud nodes. To develop a system which ensure that the security and data storage efficiency in cloud and also the integrity checking is designing effectively. Our problem is formulated as to guarantee security and data storage efficiency in cloud, integrity checking is designed effectively. Enhance the component work of the integrity checking against the service attacks and threads. End user can store the data in cloud at anytime and anywhere through internet. Third Party Auditor can check the data on cloud server at any time for security purpose and notify the data owner about stored data.

The system architecture consists of following Modules:

A. FILE FRAGMENTATON

The file fragmented means to be broken up into small pieces. This is exactly what happensto files when they become fragmented. They are broken down into small individual pieces and stored in random locations. This causes less than optimal processing times because it takes longer to read through and find all of the different locations of the file, instead of being able to look in just one location. In a fragmentation schema file f is split into n fragments, all fragments are signed and distributed to n multiples nodes, one fragment per node. The user can reconstruct file f by accessing m fragments arbitrarily chosen [8].
The size of fragmented file based on the following criteria:

F= Fragmentation of file.
Size=Total file size.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2016

NOF=Number of Fragments
SOF=Size of Fragment.
SOF=Size/NOF

B. FILE ENCRYPTION AND DECRYPTION

The file fragments are encrypted before uploading file to the cloud nodes by using Advance Encryption standard (Key-128 bit). When user wants to download the file at that file is decryption occurs and then original file is downloaded.

C. FRAGMENT ALLOCATION

The fragmentsare allocated by usingcentrality and T-coloring technique.

1. Centrality

To enhance information recovery time, the nodes are chosen taking into account the centrality measures that guarantee an enhanced access time. The centrality of a node in a chart gives the measure of the relative significance of a hub in the system.[1]

- **Betweenness Centrality**

The betweenness centrality of a node n is the number of the shortest paths, between other nodes, passing through n. Formally, the betweenness centrality of any node v in a network is given as:

$$C_b(v) = \sum_{a \neq v \neq b} \frac{\delta_{ab}(v)}{\delta_{ab}} \qquad (1)$$

Where δ_{ab} is the total number of shortest paths between a and b, and $\delta_{ab}(v)$is the number of shortest paths between a and b passing through v. The variable $C_b(v)$denotes the betweenness centrality for node v.[1]

2. T-Coloring

The fragments allocation provides the security while placing the fragments, the concept of T-coloring is used that was originally used for the channel assignment problem. This generates a non-negative random number and builds the set T starting from zero to the generated random number. The set T is used to restrict the node selection to those nodes that are at hop-distances not belonging to T. For this purpose, it assigns colors to the nodes, such that, initially, all of the nodes are given the open color. When a fragment is placed on the node, all of the nodes neighborhood nodes at a distance belonging to T are assigned close color. In this process, this loses some of the central nodes that may increase the retrieval time. But it achieves a higher security.

D. THIRD PARTY AUDITOR AND PROXY AGENT

To maintain integrity we are using the Third Party Auditor (TPA) which makes the audit ofthe stored file fragments on cloud and sent audit report to the data owner by mail. If the file is modified by attacker then TPA sends audit report as modified file to data owner and Proxy Agent which replace the modified with original content. As user requests for downloading, first merge the file fragments on cloud and then decrypt by using AES algorithm, if it is valid user he will get original file.

1.2. IMPLEMENTATION DETAILS

When data owner needs to send file on cloud server, first file is splinting into fragments using splinting algorithm and then apply SHA and AES algorithm for each fragments to generate hash value and encryption of each fragments respectively. These encrypted fragments are then sending to cloud server. These fragments are then allocated on cloud server by using fragment placement algorithm. To maintain integrity we are using the Third Party Auditor (TPA) which makes the audit of the stored file fragments on cloud and sent audit report to the data owner by mail. If the file is modified by attacker then TPA sends audit report as modified file to data owner and Proxy Agent which replace the

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2016

modified with original content. As user requests for downloading, first merge the file fragments on cloud and then decrypt by using AES algorithm, if it is valid user he will get original file.
In this philosophy we not to store the entire file at a single node yet we utilize different nodes to store the whole files. Every node contains one and only part. So there for the accomplishment of an assault on the resulting nodes will require less exertion when contrasted with the exertion on the first node. The measure of traded off information can be decreased by splitting file into different fragments and storing them on separate nodes. An effective interruption on a single or couple of nodes will just give access to a bit of information that won't not be of any hugeness. In addition, if an attacker is unverifiable about the areas of the pieces, the probability of finding pieces on all of the nodes is very low . Consider a cloud with M nodes and a File with z number of fragments .Let s be the number of successful intrusions on distinct nodes, such that s>z. The probability that s number of casualty nodescontains all of the z sites storing the file fragments (represented by P(s, z)) is given as: [1]

$$P(S,Z) = \frac{\binom{S}{Z}\binom{M-S}{S-Z}}{\binom{M}{S}} \qquad (2)$$

The above mathematical model can be solved by using following formula:-

$$\binom{n}{r} = C\,{}^{n}_{r} = \frac{n!}{r!\,(n-r)!} \qquad (3)$$

On the off chance that M = 30, s = 10, and z = 7, then P(10; 7) = 0:0046. In any case, on the off chance that we pick M = 50, s = 20, and z = 15, then P(20; 15) = 0:000046. With the increase in M, the probability of a state decreases further. In this manner, we can say that the greater the value of M , the less likely that an attacker will acquire the information record. In cloud sysem with a huge number of nodes, the probability for an attacker to acquire a lot of information diminishes essentially. However, putting every fragment once in the system will expand the information recovery time. [1]

1.3. MATHEMATICAL MODEL

The Mathematical model is in the form of set theory is given below:

Set Theory

S={s, e, X, Y,Φ}

Where,

Step1: s = Start of the program.

- Log in with System.
- Load File on Cloud Server.

Step2: e = End of the program.
Retrieve the useful file from cloud storage.

Step3: X = Input of the program.
Input of this system is any file that contains textual or image information.

Step4: Y = Output of the program.

- First select the file to upload and then send the file to cloud server and Third Party Auditor. After selecting file it runs fragmentation algorithm and makes fragments of file.
- After fragmentation we are encrypting that fragments. After encryption we assign that blocks to the cloud server nodes.
- As user requests for downloading if it is valid user he will get file as output.

X, Y ∈ U

Step5: Let U be the Set of System. U= {Client, F, TP, A, T}

Where Client, F, R, A, are the elements of the set.

1. Client=Data Owner, User
2. F= Fragmentation of file.
   - Size=Total file size.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2016

- NOF=Number of Fragments
- SOF=Size of Fragment.
  - SOF=Size/NOF

3. TP=Third Party Auditor.

- SHA=Hash value of original file stored at TPA side.
- SHA1=Hash value of original file stored at cloud.

If (SHA=SHA1)
File is safe
Else
File is not safe.

4. A= Allocating blocks to different nodes on cloud server.
5. T=T coloring graph concept to allocate fragments.

Step6: Φ = Failures and Success conditions.

FAILURES

- Huge database can lead to more time consumption to get the information.
- Hardware failure.
- Software failure.

SUCCESS

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

1.4. ALGORITHM

1.4.1. FILE SPLITTING AND MERGING

The file splitting and merging algorithm is used to split the file into different fragments before sending file over the cloud nodes. The fragmented file is storing on cloud nodes based on the concept of T-coloring and centrality. When user want to download the file from the cloud then merge that file fragments in the reverse order of file splitting.

Algorithm 1: File splitting and Merging

Input: File that contains any type of textual format
(sof: Size of fragment)
Step-1:***If*** (file is to be split) then go to step 2
Else
Merge the fragments of the file and goto step 10
Step-2: Input srcpath, destnpath, sof
Step-3: size= size of source file
Step-4: Display the size
Step-5: ***If*** (size>sof) then goto step 6
Else
Display file cannot be split and goto step 10
Step-6: Split into fragment=sof
Step-7: size=size-sof
Step-8: ***If*** (size>sof) then goto step 6

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2016

Step-9: We get fragments with merge option
Step-10: End.

1.4.2. FRAGMENT PLACEMENT

To provide the security while placing the fragments, the concept of T-coloring is used that was originally used for the channel assignment problem. This generates a non-negative random number and builds the set T starting from zero to the generated random number. The set T is used to restrict the node selection to those nodes that are at hop-distances not belonging to T. For this purpose, it assigns colors to the nodes, such that, initially, all of the nodes are given the open color. When a fragment is placed on the node, all of the nodes neighborhood nodes at a distance belonging to T are assigned close color. In this process, this loses some of the central nodes that may increase the retrieval time. But it achieves a higher security level. If anyhow the intruder compromises a node and obtains a fragment, he cannot determine the location of the other fragments. The attacker can only keep on guessing the location of the other fragments, because the nodes are separated by T-coloring. The algorithm of fragment placemat by using T-coloring is as shown below [1].
The symbols and its meaning are as shown in the following table 1.

Symbols	Meaning
O	Set of fragments
o	set of size of fragments
O_k	k-th fragment of file
o_k	Size of O_k
col	Set of colors
cen	set of centrality measure
S^i	i-th node
s_i	size of S^i
col_{S^i}	color assigned to S^i
cen_i	centrality measure for S^i
T	A set containing distance by which assignment of fragments must be separated.

Table 1: Symbols and its meaning

Algorithm 2: Fragment Placement

Inputs and initializations:
Step-1: O = { O_1, O_2,..... ,O_N }
Step-2 : o = { sizeof(O_1), sizeof(O_2), ... ,sizeof(O_N) }
Step-3: col = { open_color , close_color}
Step-4 :cen = { cen_1, cen_2, ..., cen_M}
Step-5 : col ← open_color , for All i
Step-6 :cen$_i$,for All i
Compute:-
Step- 7 :***for*** each O_k belongs to O do
Select S^i | S^iindex of(max(cen_i))
Ifcol_{S^i} =open_color and s_i> = o_k then

S^i ← O_k
$s_i s_i$ ← o_k

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2016

```
        col_S^i close_color
        S^i ` distance(S^i,T)   /* returns all node at   distance T form  S^i  and stores in temporary set S^i ‘  */
        col_S^i ` close_color
    end if
    end for
```

1.4.3. ADVANCED ENCRYPTION STANDARD

AES is a symmetric block cipher. This means that it uses the same key for both encryption and decryption. AES standard states that the algorithm can only accept a block size of 128 bits and a choice of three keys - 128, 192, 256 bits. Depending on which version is used, the name of the standard is modified to AES-128, AES-192 or AES- 256 respectively. A number of AES parameters depend on the key length. For example, if the key size used is 128 then the number of rounds is 10 whereas it is 12 and 14 for 192 and 256 bits respectively. At present the most common key size likely to be used is the 128 bit key.

Algorithm 3: Algorithm for Advanced Encryption Standard (AES-128)

One AES round applies all four operations on the state consecutively for both Encryption and Decryption[9] .

Stpe1: Encryption

```
voidAES round(unsigned char _state, unsigned char _roundKey)
        {
                subBytes(state);
                shiftRows(state);
                mixColumns(state);
                addRoundKey(state, roundKey);

        }
```

- **Byte Substitution (Sub Bytes)**
  The 16 input bytes are substituted by looking up a fixed table (S-box) given in design.The result is in a matrix of four rows and four columns.

- **Shift rows**
  Each of the four rows of the matrix is shifted to the left. Any entries that fall off arere-inserted on the right side of row. Shift is carried out as follows

a. First row is not shifted.
b. Second row is shifted one (byte) position to the left.
c. Third row is shifted two positions to the left.
d. Fourth row is shifted three positions to the left.
e. The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

- **Mix Columns**

Each column of four bytes is now transformed using a special mathematical function. This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the original column. The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2016

- **Add round key**

The 16 bytes of the matrix are now considered as 128 bits and are XOR to the 128 bitsof the round key. If this is the last round then the output is the cipher text. Otherwise,the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.

Step2: ***Decryption***

```
voidAES invRound(unsigned char _state,      unsigned char _roundKey)
{
invShiftRows(state);
invSubBytes(state);
        addRoundKey(state, roundKey);
        invMixColumns(state);
                        }
```

The process of decryption of an AES cipher text is similar to the encryption process in the reverse order.
Each round consists of the four processes conducted in the reverse order.

- Add round key
- Mix columns
- Shift rows
- Byte substitution

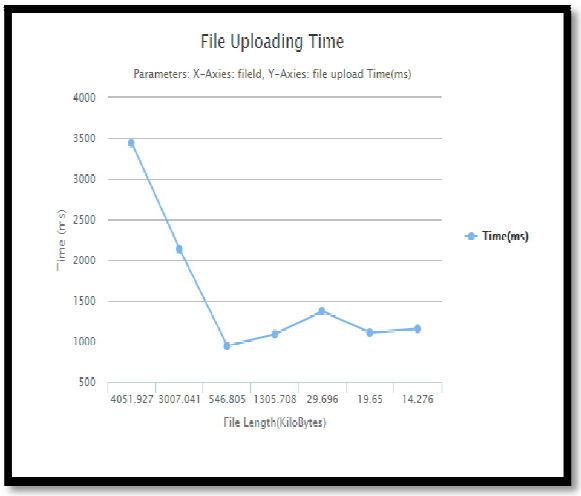
Since sub-processes in each round are in reverse manner, unlike for a cipher, the encryption and decryption algorithms need to be separately implemented, although they are very closely related.

1.5. EXPERIMENTAL SETUP

The system is built using java framework (version JDK 7) on Windows platform. The Eclipse Java EE IDE, Tomcat server and cloud server used as a development tool. The system does not require any specific hardware to run; any standard machine is capable of running the application.

IV. RESULTS & CONCLUSION

1.6. File Uploading Graph



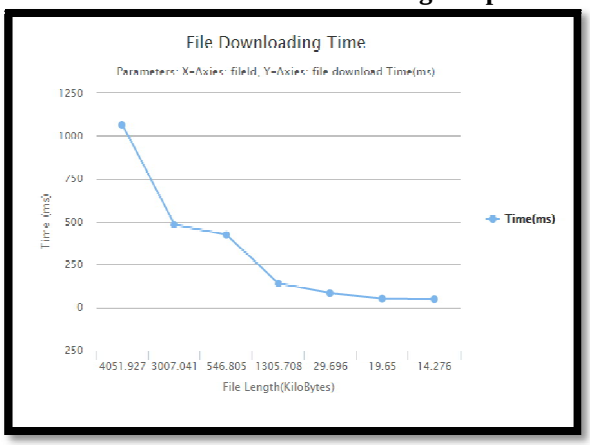
ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

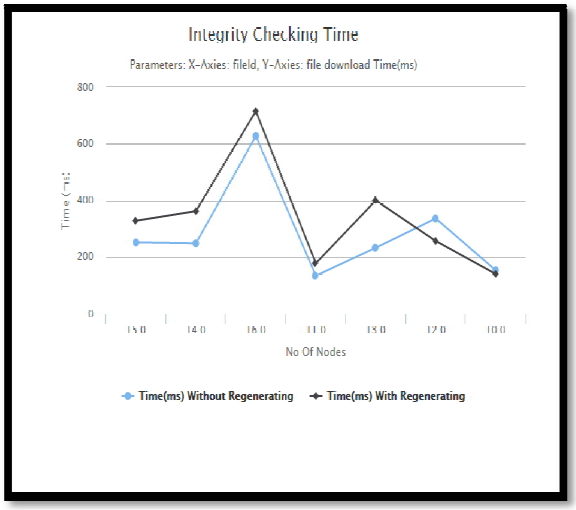
(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2016

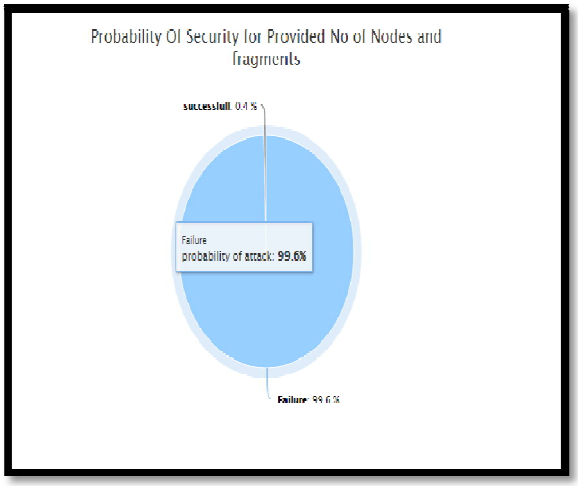
1.7. File Downloading Graph



1.8. Integrity Checking graph



1.9. Security Graph



ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2016

The propose system gives data storage security in Cloud Computing. In the proposed methodology, the data file was fragmented and the fragments are dispersed 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 single fragment of the same file.

V. FUTURE SCOPE

The proposed system uses the technique Fragmentation, T-coloring and auditing to give security and optimal performance of data in cloud. In this system only textual files are fragmented not images, video and audio file. So in future we can fragment images, audio or video files. This system uses single auditing but in future we will add for batch auditing.

REFERENCES

[1] Mazharali, Kashif Bilal, Samee U. Khan, BharadwajVeeravalli, Keqin Li, Albert Y. Zomaya "DROPS: Division and Replication of Data inCloud for Optimal Performance and Security" DOI 10.1109/TCC.2015.2400460, IEEE Transactions on Cloud computing.

[2] P. Mell and T. Grance, Draft NIST working definition of cloud computing.

[3] Jun Feng, Yu Chen, Wei-Shinn Ku, Zhou Su "D-DOG: Securing Sensitive Data in Distributed Storage Space by Data Division and Out-of-order keystream Generation" IEEE Communications Society subject matter experts for publication in the IEEE ICC 2010 proceeding.

[4] Alessandro Mei, Luigi V. Mancini, and SushilJajodia "Secure Dynamic Fragment and Replica Allocation in Large- Scale Distributed File Systems" IEEE transactions on parallel and distributed systems, vol. 14, no. 9, September 2003.

[5] Parakh A, and Kak S (2009). Online data storage using implicit security, Information Sciences, vol 179(19), 3323–3331.

[6] CongWang, Sherman S.-M. Chow, QianWang, KuiRenandWenjing Lou Privacy-Preserving Public Auditing for Secure Cloud Storage IEEE paper to support US National Science Foundation under grant CNS-0831963, CNS- 0626601, CNS-0716306, and CNS-0831628.

[7] D.H. Patil Rakesh R. Bhavsar and Akshay S. Thorve Data Security over Cloud Emerging Trends in Computer Science and Information Technology 2012 (ETCSIT2012) Proceedings published in International Journal of Computer Applications (IJCA). J. Cryptology, vol. 21,no. 4, pp. 469-491, 2008.

[8] A. Mei, L. V. Mancini, and S. Jajodia, Secure dynamic fragment and replica allocation in large-scale distributed file systems, IEEE Transactions on Parallel and Distributed Systems, Vol. 14, No. 9, 2003, pp. 885-896.

[9] LIU Niansheng, GUO Donghai and HUANG jiaxiang, AES algorithm implemented for PDA secure communication with java, in 2007 IEEE ,DOI:1-4244- 1035, vol 5,july 2007.

[10] Deshmukh P M, Gughane A S et al. (2012). Maintaining File Storage Security in Cloud Computing, International Journal of Emerging Technology and Advanced Engineering, vol 2(10),22502459.

.