Parting and Replication of Data for Performance and Security Optimization in Cloud

[1]Harini Priya P, [2]Dr. Paras Nath Singh
[1]M Tech(CSE), [2]Professor(CSE)
Department of Computer Science and Engineering,
CMR Institute of Technology, Bangalore, India

Abstract: In cloud computing Redistributing data to an outsider managerial control, offers ascend to security concerns. The data bargain may happen because of attacks by different users or nodes inside the cloud. Subsequently, high safety efforts are required to secure data inside the cloud. Be that as it may, the utilized security methodology should likewise consider the improvement of the data recovery time. In this paper, we propose Parting and Replication of Data for Performance and Security optimization in cloud (PRPSO) to handle the performance and security issues. In this approach, we fragment data files which are uploaded by data owner, and replication over the cloud nodes are performed. Every one of the node stores just a solitary piece of a specific data file in encrypted form that guarantees that even in an occurrence of an effective attack on a node, the data can be retrieved.

IndexTerms **- Cloud, AES, Fragmentation, Replication, Performance, Security, Optimization.**

I. INTRODUCTION

In this Internet era storing of data in manual scripts is not a wise thing, all know the importance of data storage and security. Dealing with data is became part of our day to day life. Storing of data is being done by Internet although some other transactions also being done by internet most of it is roam around data so when come to security of data it will not provide sufficient Security. We need to opt another sophisticated way to store data, one among is cloud computing. Cloud computing provides scalable computing and storage resources via the Internet. It also enables users to access services without regard to where the services are provided and how they are delivered. Cloud is mainly used for the public use and is said to be public cloud, where at public cloud the data being outsourced to third party. So, there comes the issue of the security which is to be concentrated. Security is one of the most crucial aspects among those prohibiting the wide-spread adoption of cloud computing.

So, in this methodology, we partition the data uploaded by data owner and encrypt using AES algorithm, and replicate them at distinct locations within the cloud. Replication is performed to improve retrieval time, progress system availability (by directing traffic to a replica after a failure), avoid data loss (by recovering lost data from a replica), and improve performance (by spreading load across multiple replicas and by making low-latency access available to users around the world). Every one of the node stores just a solitary piece of a specific data file in encrypted form to improve data security. In an occurrence of an effective attack on a node, data can be retrieved from the replicas.

The rest of the paper is organized as follows section II deals with Literature Survey, section III speaks about Proposed Methodology, section IV speaks about Results, section V and VI speaks about the Conclusion and Future Work.

II. LITERATURE SURVEY

Mazhar AliKashif Bilal et.al [9], were the first to present DROPS Division and Replication of Data in Cloud for Optimal Performance and Security Methodology, where the information to be put away into the cloud and to be gotten to by the other user is being separated and recreated over the cloud from fundamental node to the sub nodes and keeping in mind that downloading data the way creation is produced using sub nodes to principle node which beats the cryptographic method for getting to the information and an ideal exhibition is picked up as information is isolated.

Juels [7], introduced a procedure to guarantee the respectability, freshness, and accessibility of information in a cloud. The information relocation to the cloud is performed by the Iris document framework. An entryway application is planned and utilized in the association that guarantees the respectability and freshness of the information utilizing a Merkle tree. The document squares, MAC codes, and form numbers are put away at different dimensions of the tree. The proposed system in [7] intensely relies upon the users utilized plan for information classification. In addition, the plausible measure of misfortune if there should arise an occurrence of information treating because of interruption or access by different VMs can't be diminished.

G.Kappes [8], approached multi-tenant issues with the usage of consolidated storage and local access control. The proposed framework works for object based file systems. A trusted third-party is used to improve trust in the authentication.

Deswarte [10], discussed about data replication, which conveys data closer to data purchasers, is viewed as a promising arrangement. It permits limiting system deferrals and data transfer capacity utilization.

Carlson [12], proposed an investigation over cloud computing for the issues of security and have proposed arrangement inside that. The proposed arrangement over security worried about cloud computing information.

Varun [1], implemented the old philosophy of DROPS Methodology where just the division of the data is being taken among nodes and the duplication of the information isn't stayed away from which influences the cloud to devour more space and even it takes more expense. Giving the de-duplication of the information to be put away on the cloud by the assistance of two systems, for example, record level and node level de-duplication strategies.

III. PROPOSED METHODOLOGY

In the event that in a cloud, the entire file is put away on a solitary node then fruitful attack on that node will put the security of entire file in danger. In these frameworks, the replication time can be improved by reproducing the data at different nodes. In any case, this outcome in increment in attack surface, and along these lines increment in danger of data security on numerous occasions. So in these frameworks expanded performance results in diminished security. In our approach the entire file isn't put away on a solitary node; division of file into different parts is done and put away on various nodes. Cloud server (a user facing server in the cloud that entertains user's requests) performs following functions:

- Receiving the file
- Encryption of file by AES algorithm
- Fragmentation (stores one fragment over each of the selected node)
- replication (The cloud manager keeps record of the fragment placement)

In our proposed system, the data owner uploads his file which is then partitioned using fragment placement algorithm. This partitioned data is encrypted using AES algorithm before uploading to the cloud server. After successful upload the cloud server keeps track of the details of data owners, end users, files uploaded, throughput, time delay results and replication results. When the end user need to download a particular file he has to send request to the cloud server. After the request is approved by the cloud server the end user will get the encrypted data which is decrypted using AES algorithm and then the data file can be downloaded.

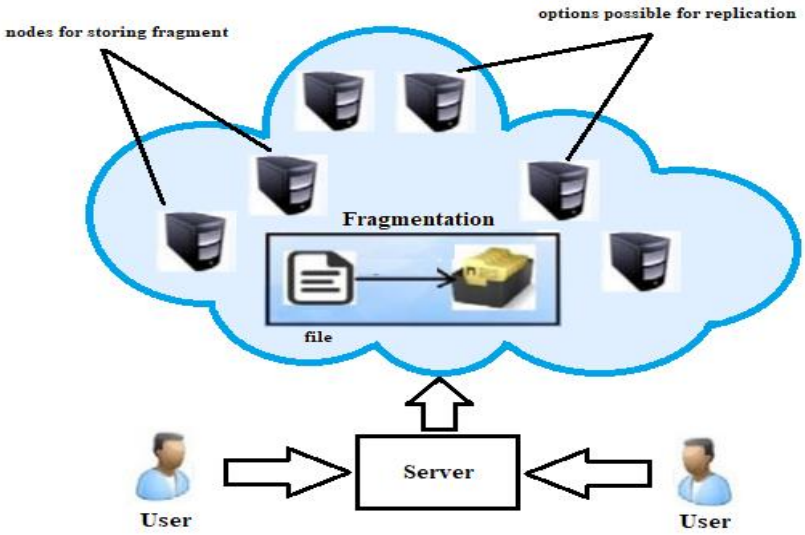

Fig 1: Architecture of proposed methodology

3.1 ALGORITHM 1: Advanced Encryption Standard (AES)

Encryption Process:
i) Key Expansions—Round keys are derived from the cipher key using Rijndael's key schedule. A different 128-bit round key for each round is required.
ii) Sub Bytes—16 input bytes are substituted by looking up a fixed table, result is a matrix of 4 rows and 4 columns.
iii) Shift Rows—A transposition step where the last three rows of the state are shifted cyclically a certain number of steps, first row is not shifted. Results in a unique matrix of the same 16 bytes with shifts.
iv) Mix Columns—This function takes the 4 bytes as input of one column and outputs 4 completely new bytes. This step is not performed in the last round.
v) Add Round Key—The 16 bytes of the matrix are now considered as 128 bits and are XORed to the 128bits of the round key.
vi) Final Round—Mix Columns isn't carried out in this step and the output is ciphertext.
Decryption process: Carried out in the opposite direction of the above encryption process.

3.2 ALGORITHM 2: Fragment Placement

Inputs and initialization:
$O = \{ O_1, O_2, \ldots\ldots, O_N \}$
$o = \{ sizeof(O_1), sizeof(O_2), \ldots.. sizeof(O_N) \}$
Col = {open_color,close_color}
$Cen = \{cen_1, cen_2 \ldots\ldots, cen_M\}$
Col ←open_color ∀ i
Cen ← cen_i ∀ i
Compute:
for each $O_k \in O$ do
Select S^i |S^i ← 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)
ColSi' ← close_color
end if
end for

Fragmentation: Security of every node decides cloud security. Attack on one node makes simple for the interloper to attack on consequent nodes in the event of homogenous framework. Yet, in heterogeneous framework same exertion won't result in interruption of ensuing nodes. In this framework attack on one node results in traded off security of data accessible just on that node, on the grounds that in this framework data file is divided and put away on various nodes. In the event that there are a large number of nodes in a cloud framework, at that point that cloud framework is generally progressively secure.

3.3 ALGORITHM 3: Fragment Replication

For each O_k in O do
select S^i that has max ($R^i_k + W^i_k$)
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
$S^i' \leftarrow$ mdistance(S^i,T)
$col_S^i \leftarrow$ close_color
End if
end for

Replication: After fragmentation and nodal position, parts of file are repeated. Replication is in a controlled way. With the goal that just a single imitation of each piece is made. Because of replication the entrance time is diminished consequently execution is expanded.

3.4 MODULES:

i) Data Owner Module

In this module, the data owner uploads their data in the cloud server. For the security purpose the data owner encrypts the data file partition and then store in the cloud. The data owner can check the replication of the file blocks over Corresponding cloud server. The Data owner can have capable of manipulating the encrypted data file fragments and the data owner can check the cloud data as well as the replication of the specific file fragments and also he can create remote user with respect to registered cloud servers. The data owner also checks data integrity proof on which the block is modified by the attacker.

ii) Cloud Server Module

The cloud service provider manages a cloud to provide data storage service. Data owners encrypt their data file fragments and store them in the cloud for sharing with Remote User. To access the shared data file fragments, data consumers download encrypted data file from the cloud and then decrypt them.

iii) End User Module

In this module, remote user logs in by using his user name and password. After he logs in he tries to download file by entering file name from cloud server.

iv) Data Encryption and Decryption

All the legal users in the system can freely query any interested encrypted and decrypted data. Upon receiving the data from the server, the user runs the decryption algorithm Decrypt to decrypt the cipher text by using its secret keys from Users.

v) Attacker Module

The user who attacks or modifies the block content called attacker. Here the attacker just modifies the data of a specific fragment. When the data owner checks for fragment safety he will be indicated and data is retrieved from the replicas.

IV. RESULTS

In PRPSO methodology the data file is partitioned into fragments and encryption is performed. The encrypted fragments are uploaded to the cloud. The cloud server calculates throughput, time delay results and replication results. When the end user requests for a file cloud server provides the authority to download. The end user receives encrypted file which is decrypted before downloading. When a particular node is attacked, using replicated node the data is retrieved successfully.

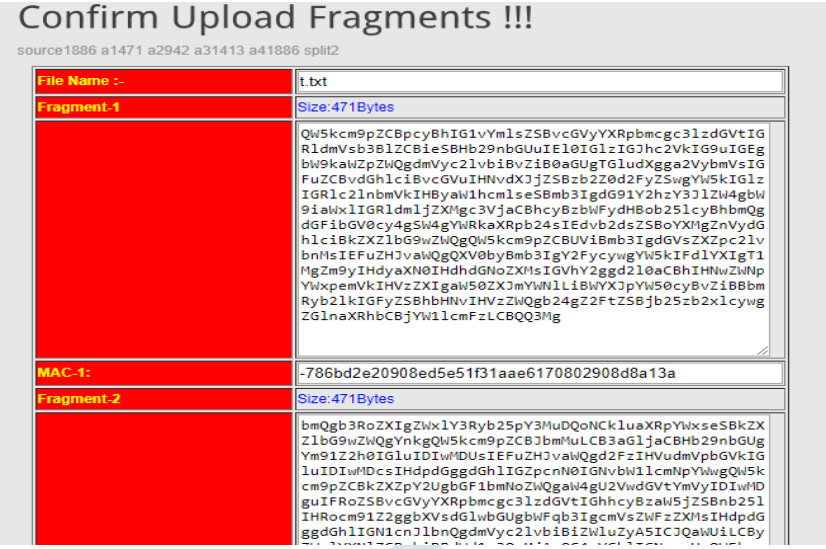

Fig 2: Partitioning of data file

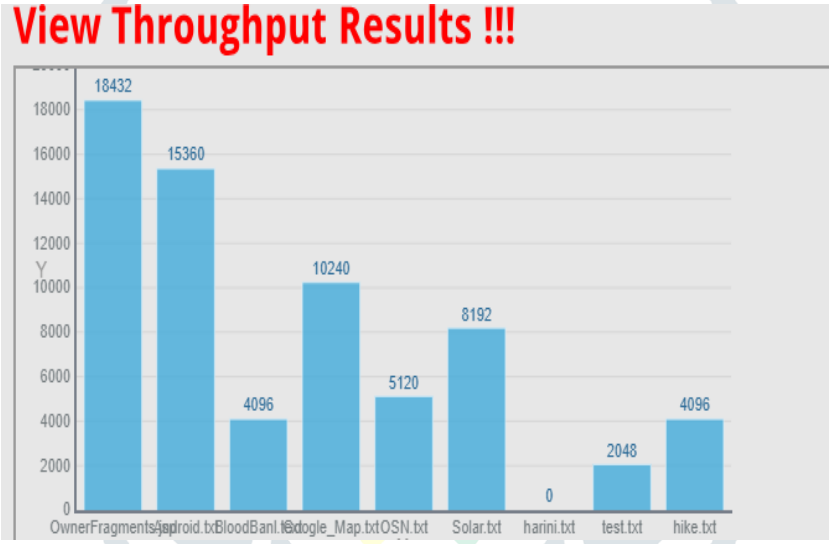

Fig 3: Throughput results from the cloud server

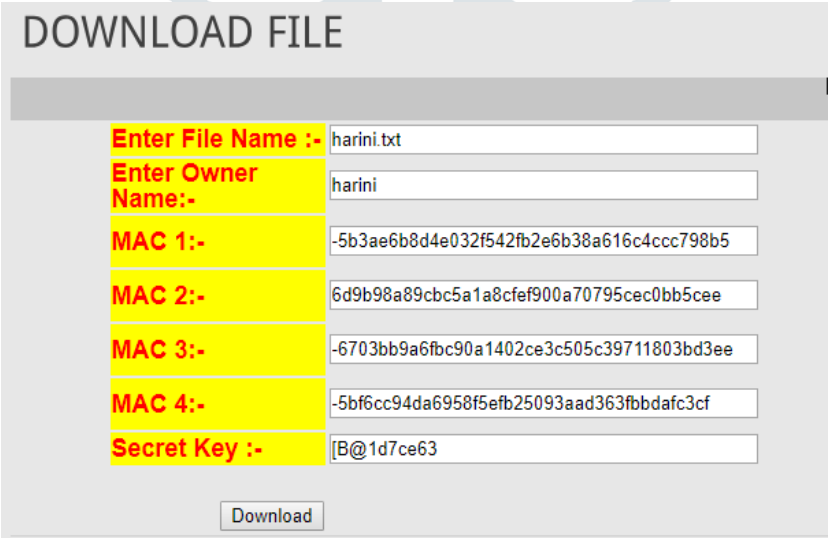

Fig 4: File details before download from the end user

V. CONCLUSION

We proposed the PRPSO approach, a cloud storage security conspire that all things considered arrangements with the security and performance. The data file was divided and the pieces are scattered over numerous nodes. The discontinuity and dispersal guaranteed that if there should arise an occurrence of a fruitful attack, data is recovered. No node in the cloud, stored more than a single fragment of the equivalent file. Throughput, time delay and replication results are given by the cloud server. Our proposed strategy brought about expanded security dimension of data.

VI. FUTURE WORK

In future work, we can stretch out the project to actualize different replica management systems and furthermore execute in mobile cloud condition. And furthermore decrease the transfer speed and throughput at the season of data sharing.

REFERENCES

[1] SRICSO: Section and Replication of Information in Cloud for Security and Optimum Operation Varun A H, S. Pramela Devi Volume 06, Issue 05, May 2017.

[2] Increasing Security and Performance of Cloud Storage Using Data Division and Replication Strategies Suyog Ghodey, Niranjan Dandekar, Sagar Kate, Vikrant Bhalerao, Shubham Bhange, Alka Londhe, Vol No. 6, December 2017.

[3] Optimization of Performance and Security by Division and Replication of Data in Cloud Gandepalli Tanuja, Vol No.3, December 2016.

[4] CONSISTENCY AND PRIVACY BASED REPLICATION SYSTEM IN CLOUD SHARING FRAMEWORK, N. Santhi & R. Selva Kumar, Volume I, Issue I, 2016.

[5] Security Enhancement of Data in Cloud using Fragmentation and Replication, P.D. Patni, Dr. S.N.Kakarwal, Volume-6, Issue-5, September-October 2016.

[6] Ms. A. Sivasankari, Ms. D. Abirami, and Mrs. K. Ayesha, "Division and replication of data in cloud for optimal performance and security using fragment placement algorithm," International Research Journal of Advanced Engineering and Science, Volume 1, Issue 4, pp. 57-63, 2016.

[7] 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.

[8] 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.

[9] DROPS: Division and Replication of Data in Cloud for Optimal Performance and Security Mazhar Ali, Student Member, IEEE, Kashif Bilal, Student Member, IEEE, Samee U. Khan, Senior Member, IEEE, Bharadwaj Veeravalli, Senior Member, IEEE, Keqin Li, Senior Member, IEEE, and Albert Y. Zomaya, Fellow, IEEE.

[10] Y. Deswarte, L.Blain, and J-C. Fabre, "Intrusion tolerance in distributed computing systems," In Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy, Oakland CA.

[11] Apsdrdo: Adaptive Particle Swarm Division and Replication of Data Optimization for Security in Cloud Computing, P.Jayasree, Dr.V.Saravanan.

[12] Frederick R. Carlson Saint Petersburg College Saint Petersburg, Florida 352-586-2621 "Security Analysis of Cloud Computing"fcarlson@ieee.org.