International Journal of Innovative Research in Science and Engineering
Vol. No.3, Issue 04, April 2017
www.ijirse.com
ISSN (O) 2454-9665
ISSN (P) 2454-0663

ADVANCE SECURITY FOR DYNAMIC DATA IN CLOUD ENVIRONMENT

Khade Harshada, Salunkhe Pritish, Kulkarni Pooja, Kadam Rohan

GenbaSopanraoMoze College of Engineering,

SavitribaiPhule Pune University, Pune, India

ABSTRACT

Nowadays, more and more enterprises and organizations are hosting their data into the cloud, in order to reduce the IT maintenance cost and enhance the data reliability. The general status quo is that customers usually put their data into a single cloud (which is subject to the vendor lock-in risk) and then simply trust to luck. Outsourcing data to a third-party administrative control, as is done in cloud computing, gives rise to security concerns. The data compromise may occur due to attacks by other users and nodes within the cloud. Therefore, high security measures are required to protect data within the cloud. However, the employed security strategy must also take into account the optimization of the data retrieval time. In this methodology, we divide a file into fragments, and allocate the fragmented data over the cloud nodes. Each of the nodes stores only a single fragment of a particular data file that ensures that even in case of a successful attack, no meaningful information is revealed to the attacker. Moreover, the nodes storing the fragments, are separated with certain distance by means of graph T-coloring to prohibit an attacker of guessing the locations of the fragments. We also propose a public auditing scheme for the regenerating-code-based cloud storage. To solve the regeneration problem of failed authenticators in the absence of data owners, we introduce a proxy, which is privileged to regenerate the authenticators, into the traditional public auditing system model. Thus, our scheme can completely release data owners from online burden.

Keywords: Cloud storage,CloudSecurity,Fragmentation, Public audit

I. INTRODUCTION

From the existing work survey, we can deduce that both security and performance are critical for the next generation large-scale systems, such as clouds. Therefore, in this project, we collectively approach the issue of security and performance as a secure data replication problem. We present Division of Data in the Cloud for Optimal Performance and Security that judicially fragments user files into pieces and allocates them at strategic locations within the cloud. The division of a file into fragments is performed based on a given user criteria such that the individual fragments do not contain any meaningful information. Each of the cloud nodes (we use the term node to represent computing, storage, physical, and virtual machines) contains a distinct fragment to increase the data security. A successful attack on a single node must not reveal the locations of other fragments within the cloud. To keep an attacker uncertain about the locations of the file fragments and to further improve the security, we select the nodes in a manner that they are not adjacent and are at certain distance from each

ISSN (O) 2454-9665
ISSN (P) 2454-0663

other. The node separation is ensured by the means of the T-coloring. To improve data retrieval time, the nodes are selected based on the centrality measures that ensure an improved access time.

II. PROPOSED SYSTEM

The data outsourced to a public cloud must be secured. Unauthorized data access by other users and processes (whether accidental or deliberate) must be prevented [14]. As discussed above, any weak entity can put the whole cloud at risk. In such a scenario, the security mechanism must substantially increase an attacker's effort to retrieve a reasonable amount of data even after a successful intrusion in 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 [15]. A key factor determining the throughput of a cloud that stores data is the data retrieval time [21]. In large-scale systems, the problems of data reliability, data availability, and response time are dealt with data replication strategies [3]. However, placing replicas data over a number of nodes increases the attack surface for that particular data. For instance, storing m replicas of a file in a cloud instead of one replica increases the probability of a node holding file to be chosen as attack victim, from 1 n to m n , where n is the total number of nodes.

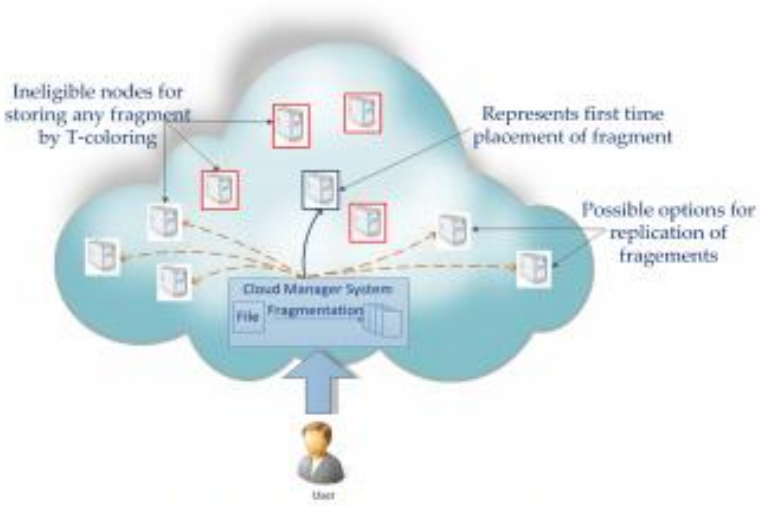


Fig. 1: The DROPS methodology

III. DATA FRAGMENTATION

The security of a large-scale system, such as cloud depends on the security of the system as a whole and the security of individual nodes. A successful intrusion into a single node may have severe consequences, not only for data and applications on the victim node, but also for the other nodes. The data on the victim node may be revealed fully because of the presence of the whole file [17]. A successful intrusion may be a result of some software or administrative vulnerability [17]. In case of homogenous systems, the same flaw can be utilized to target other nodes within the system. The success of an attack on the subsequent nodes will require less effort as compared to the effort on the first node. Comparatively, more effort is required for heterogeneous systems. However, compromising a single file will require the effort to penetrate only a single node. The amount of

ISSN (O) 2454-9665
ISSN (P) 2454-0663

compromised data can be reduced by making fragments of a data file and storing them on separate nodes [17, 21]. A successful intrusion on a single or few nodes will only provide access to a portion of data that might not be of any significance. Moreover, if an attacker is uncertain about the locations of the fragments, the probability of finding fragments on all of the nodes is very low. Let us 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 victim nodes contain all of the z sites storing the file fragments (represented by P(s,z)) is given as: $P(s, z) = (s z) (M - s s - z) (M s)$.

3.1 T-coloring

Suppose we have a graph G = (V, E) and a set T containing non-negative integers including 0. The Tcoloring is a mapping function f from the vertices of V to the set of non-negative integers, such that $|f(x)- f(y)| \notin T$, where $(x, y) \in E$. The mapping function f assigns a color to a vertex. In simple words, the distance between the colors of the adjacent vertices must not belong to T. Formulated by Hale [6], the T-coloring problem for channel assignment assigns channels to the nodes, such that the channels are separated by a distance to avoid interference.

3.2 Comparative Techniques

We compared the results of the DROPS methodology with fine-grained replication strategies, namely: (a) DRPA-star, (b) WA-star, (c) A-star, (d) SA1, (e) SA2, (f) SA3, (g) Local Min-Min, (h) Global MinMin, (i) Greedy algorithm, and (j) Genetic Replication Algorithm (GRA). The DRPA-star is a data replication algorithm based on the A-star best-first search algorithm. The DRPA-star starts from the null solution that is called a root node. The communication cost at each node n is computed as: cost(n) = g(n) + h(n), where g(n) is the path cost for reaching n and h(n) is called the heuristic cost and is the estimate of cost from n to the goal node. The DRPA-star searches all of the solutions of allocating a fragment to a node. The solution that minimizes the cost within the constraints is explored while others are discarded. The selected solution is inserted into a list calledthe OPEN list. The list is ordered in the ascending order so that the solution with the minimum cost is expanded first. The heuristic used by the DRPAstar is given as h(n) = max(0, (mmk(n)g(n))), where mmk(n) is the least cost replica allocation or the maxmin RC. Readers are encouraged to see the details about DRPA-star in [13]. The WA-Star is a refinement of the DRPA-star that implements a weighted function to evaluate the cost. The function is given as: f(n) = f(n) + h(n) + (1 − (d(n)/D)h(n). The variable d(n) represents the depth of the node n and D denotes the expected depth of the goal node [13]. The A-star is also a variation of the DRPA-star that uses two lists, OPEN and FOCAL. The FOCAL list contains only those nodes from the OPEN list that have f greater than or equal to the lowest f by a factor of 1 + . The node expansion is performed from the FOCAL list instead of the OPEN list

3.3 Workload

The size of files were generated using a uniform distribution between 10Kb and 60 Kb. The primary nodes were randomly selected for replication algorithms. For the DROPS methodology, the S i′ s selected during the first cycle of the nodes selection by Algorithm 1 were considered as the primary nodes. The capacity of a node was generated using a uniform distribution between (1 2 CS)C and (3 2 CS)C, where $0 \leq C \geq 1$. For instance, for CS = 150 and C = 0.6 the capacities of the nodes were uniformly distributed between 45 and 135. The mean

ISSN (O) 2454-9665
ISSN (P) 2454-0663

value of g in the OPEN and FOCAL lists was selected as the value of , for WA-star and A-star, respectively. The value for level R was set to ⌊ d 2 ⌋, where d is the depth of the search tree(number of fragments). The read/write (R/W) ratio for the simulations that used fixed value was selected to be 0.25 (The R/W ratio reflecting 25% reads and 75% writes within the cloud). The reason for choosing a high workload (lower percentage of reads and higher percentage of writes) was to evaluate the performance of the techniques under extreme cases. The simulations that studied the impact of change in the R/W ratio used various workloads in terms of R/W ratios. The R/W ratios selected were in the range of 0.10 to 0.90. The selected range covered the effect of high, medium, and low workloads with respect to the R/W ratio

3.4 Results and Discussion

We compared the performance of the DROPS methodology with the algorithms discussed in Section 5.1. The behavior of the algorithms was studied by: (a) increasing the number of nodes in the system, (b) increasing the number of objects keeping number of nodes constant, (c) changing the nodes storage capacity, and (d) varying the read/write ratio. The aforesaid parameters are significant as they affect the problem size and the performance of algorithms [13].

FIGURES AND TABLES

TABLE 1: Notations and their meanings

Symbols	Meanings
M	Total number of nodes in the cloud
N	Total number of file fragments to be placed
O_k	k-th fragment of file
o_k	Size of O_k
S^i	i-th node
s_i	Size of S^i
cen_i	Centrality measure for S^i
col_{S^i}	Color assigned to S^i
T	A set containing distances by which assignment of fragments must be separated
r_k^i	Number of reads for O_k from S^i
R_k^i	Aggregate read cost of r_k^i
w_k^i	Number of writes for O_k from S^i
W_k^i	Aggregate write cost of w_k^i
NN_k^i	Nearest neighbor of S^i holding O_k
$c(i,j)$	Communication cost between S^i and S^j
P_k	Primary node for O_k
R_k	Replication schema of O_k
RT	Replication time

International Journal of Innovative Research in Science and Engineering
Vol. No.3, Issue 04, April 2017
www.ijirse.com
ISSN (O) 2454-9665
ISSN (P) 2454-0663

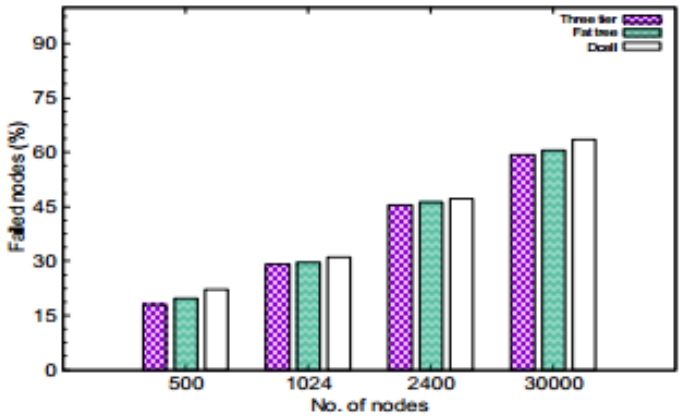
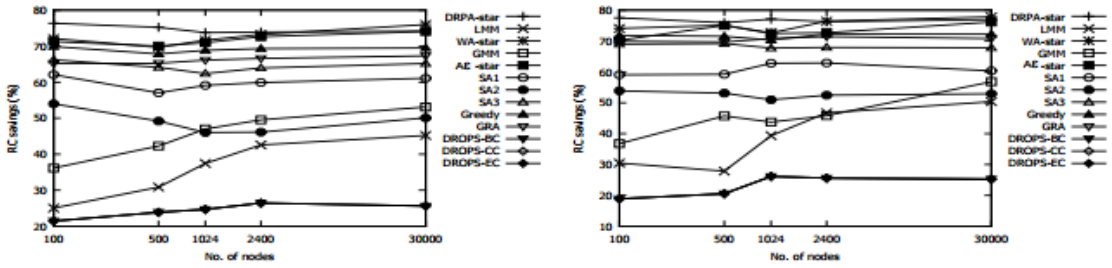
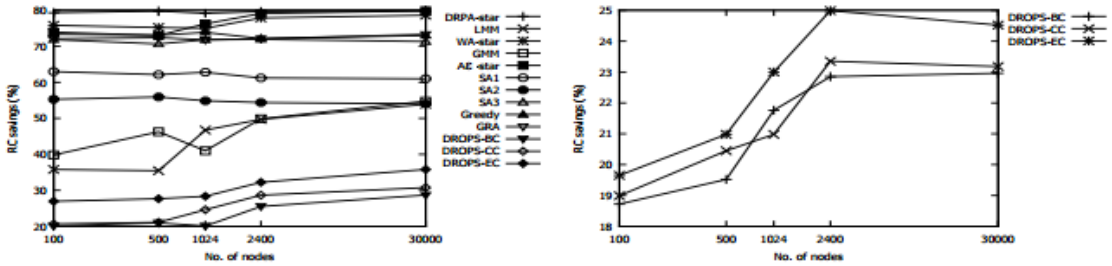


Fig. 10: Fault tolerance level of DROPS



(a) (b)

Fig. 2: (a) RC versus number of nodes (Three tier) (b) RC versus number of nodes (Fat tier)



(a) (b)

Fig. 3: (a) RC versus number of nodes (Dcell) (b) RC versus number of nodes for DROPS variations with maximum available capacity constraint (Three tier)

IV. CONCLUSION

We proposed the DROPS 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 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. The performance of the DROPS 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. Currently with the DROPS methodology, a user has to download the file, update the

ISSN (O) 2454-9665
ISSN (P) 2454-0663

contents, and upload it again. It is strategic to develop an automatic update mechanism that can identify and update the required fragments only. The aforesaid future work will savethe time and resources utilized in downloading, updating, and uploading the file again. Moreover, the implications of TCP incast over the DROPS methodology need to be studied that is relevant to distributed data storage and access.

V. ACKNOWLEDGEMENTS

We compared the performance of the DROPS methodology with the algorithms discussed in Section 5.1. The behavior of the algorithms was studied by: (a) increasing the number of nodes in the system, (b) increasing the number of objects keeping number of nodes constant, (c) changing the nodes storage capacity, and (d) varying the read/write ratio. The aforesaid parameters are significant as they affect the problem size and the performance of algorithms [13].

Impact of increase in number of cloud nodes We studied the performance of the placement techniques and the DROPS methodology by increasing the number of nodes. The performance was studied for the three discussed cloud architectures. The numbers of nodes selected for the simulations were 100, 500, 1,024, 2,400, and 30,000. The number of nodes in the Dcell architecture increases exponentially [2]. For a Dcell architecture, with two nodes in the Dcell0, the architecture consists of 2,400 nodes. However, increasing a single node in the Dcell0, the total nodes increases to 30, 000 [2]. The number of file fragments

REFERENCES

[1.] 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

[2.]] 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.

[3.] 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.

[4.] 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, pp. 110-121, 1991.

[5.] M. Hogan, F. Liu, A.Sokol, and J. Tong, "NIST cloud computing standards roadmap," NIST Special Publication, July 2011.

[6.]] 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

[7.] A. N. Khan, M.L. M. Kiah, S. A. Madani, and M. Ali, "Enhanced dynamic credential generation scheme for protection of user identity in mobile-cloud computing, The Journal of Supercomputing, Vol. 66, No. 3, 2013, pp. 1687-1706 .

ISSN (O) 2454-9665
ISSN (P) 2454-0663

[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.] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," Future Generation Computer Systems, Vol. 28, No. 3, 2012, pp. 583-592

[10.] M. Newman, Networks: An introduction, Oxford University Press, 2009.