International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available online http://www.ijoer.in

Vol.3., Issue.6., 2015
(Nov.-Dec.,)

REVIEW ARTICLE

ISSN: 2321-7758

A NOVEL APPROACH TO SECURITY THREATS AND COST EFFICIENT DATA HOSTING OF CLOUD DATA

SUJATA SALUNKHE[1], DHANSHRI PATIL[2]

[1]M.E.Student Nutan Maharashtra Institute of Engg & Technology, Savitribai Phule Pune University, Talegaon Dabhade ,Pune

[2]Professor, Nutan Maharashtra Institute of Engg.& Technology, Savitribai Phule Pune University, Talegaon Dabhade, Pune

SUJATA SALUNKHE

ABSTRACT

Today, many enterprises and organizations are hosting their data into the cloud, in order to reduce the IT maintenance cost and enhance the data reliability. But, facing the numerous cloud vendors as well as their different pricing policies, customers may be confused which cloud(s) are suitable for storing their data and what hosting strategy is cheaper. Generally customers put their data into a single cloud (which is subject to the vendor lock-in risk) and then simply trust to luck. The first technique is about selecting several suitable clouds and an suitable redundancy strategy to store data with minimized cost and guaranteed availability. Outsourcing data in cloud computing, gives rise to security concerns. 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. For this purpose DROPS Methodology is used. In this methodology, it divides a file into fragments, and replicates 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 placed with a certain distance by means of graph T-coloring to restrict an attacker of guessing the locations of the fragments.

Key Words: CHARM, DROPS, Fragmentation, T-coloring

I. INTRODUCTION

Existing clouds belongs to great differences in terms of both working performances and pricing policies. So different cloud vendors have their respective infrastructures and keep on upgrading them with newly emerging technology. They also design different system architectures and apply various techniques to make their services competitive. Such system designs leads to performance variations across cloud vendors. Moreover, pricing policies of existing storage services provided by different cloud vendors are different in both pricing levels and charging items. For example, Rack space does not charge for Web operations (typically via a series of Restful APIs), Google Cloud Storage charges according to bandwidth consumption, while Amazon S3 charges according to storage space. CHARM is the emerging

technique for data hosting which suggest the user the appropriate cloud vendor for his data[1]. Security is one of the most important aspects among those wide-spread adoption of cloud computing. Cloud security issues are there due to the core technology's implementation (virtual machine (VM) escape, session riding, etc.), cloud service offerings (structured query language injection, weak authentication schemes, etc.), and arising from cloud characteristics (data recovery vulnerability, Internet protocol vulnerability, etc.). For a cloud to be secure, all of the participating entities must be secure. The highest level of the system's security is equal to the security level of the weakest entity. Therefore, in a cloud, the security of data does not solely depend on an individual's security measures. The neighboring entities are also responsible to provide an opportunity to an attacker to tackle the user's defenses. The data outsourced to a public cloud must be secured. Unauthorized data access by other users and processes whether it may be accidental or deliberate must be protected. 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[2]. Moreover, the amount of loss (as a result of data leakage) must also be minimized. This will also focus on the integrity verification problem in regenerating-code-based cloud storage, especially with the functional repair strategy.

II.CHARM Overview:

Cost-efficient data Hosting scheme with high Availability in heterogeneous Multi-cloud, named CHARM. It is a novel, efficient, and heuristic-based data hosting scheme for heterogeneous multi-cloud environments. CHARM accommodates different pricing strategies, availability requirements, and data access patterns. It selects suitable clouds and an appropriate redundancy strategy to store data with minimized cost and guaranteed availability. It keeps monitoring the variations of pricing policies and data access patterns, and adaptively triggers the transition process between different data storage modes.

The architecture of CHARM is shown in Figure 1.There are four main components in CHARM: Data Hosting, Storage Mode Switching (SMS), Workload Statistic, and Predictor.

Workload Statistic keeps collecting and monitoring access logs to guide the placement of data. It also sends statistic information to Predictor which guides the action of SMS. Data Hosting stores data using replication or erasure coding, according to the size and access frequency of the data. SMS works as decision maker, whether the storage mode of certain data should be changed from replication to erasure coding or in reverse, depending on the output of Predictor. The implementation of changing storage mode runs in the background, in order not to impact online service. Predictor predicts the future access frequency of files. The time interval for prediction is one month, that is, it uses the former months to predict access frequency of files in the next month. Moreover, a very simple predictor, which uses the weighted moving average approach, works well in the data hosting model. Data Hosting and SMS are two important modules in CHARM. Data Hosting decides storage mode and the clouds that the data should be stored in.

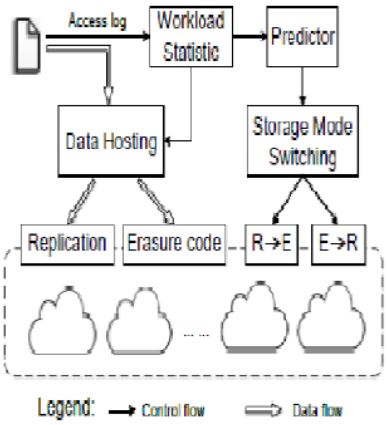

Fig.1 CHARM Architecture

III. Drops Overview:

The DROPS methodology proposes not to store the entire file at a single node. The DROPS methodology fragments the file and makes use of the cloud for replication. The fragments are distributed such that no node in a cloud holds more than a single fragment, so that even in a successful attack on the node leaks no significant information. The DROPS methodology uses controlled replication. Each of the fragments is replicated only once in the

cloud to improve the security. Although, the controlled replication does not improve the retrieval time to the level of full-scale replication, it significantly improves the security. In the DROPS methodology, user sends the data file to cloud. Upon receiving the file the cloud manager (a user facing server in the cloud that entertains user's requests) performs:

(1) Fragmentation, (2) Nodes selection and stores one fragment over each of the selected node, and (c) Nodes selection for fragments replication. The cloud manager keeps record of the fragment placement and is assumed to be a secure entity. The DROPS methodology is shown in fig.2.

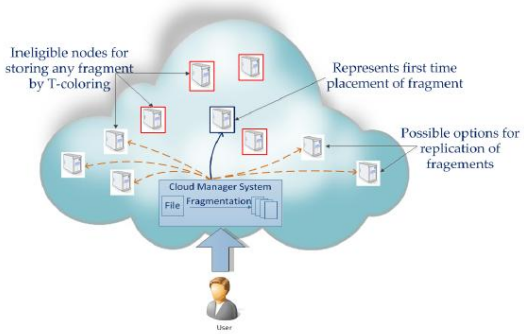

Fig.2. DROPS Methodology

A. Drops Implementation:

a) 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. A successful intrusion may be a result of some software or administrative vulnerability. The file owner specifies the fragmentation threshold of the data file is specified to be generated by. The file owner can specify the fragmentation threshold in terms of either percentage or the number and size of different fragments. The percentage fragmentation threshold, for example, can dictate that each fragment will be of 5% size of the total size of the file. Alternatively, the owner can generate separate file containing information about the fragment number and size, for instance, fragment 1 of size 4,000 Bytes, fragment 2 of size 6,749 Bytes. The owner of the file is the best candidate to generate fragmentation threshold as he is very well aware about the significant information from the file. The owner can best split the file such that each fragment does not contain significant amount of information. The default percentage fragmentation threshold can be made a part of the Service Level Agreement (SLA), if the user does not specify the fragmentation threshold while uploading the data file.

b) 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.

c) Replication : To increase the data availability, reliability, and improve data retrieval time, it also performs a controlled replication. It places the fragment on the node that provides the decreased access cost with an objective to improve retrieval time for accessing the fragments for reconstruction of original file. While replicating the fragment, the separation of fragments in the placement technique through T-coloring, is also taken care of. In case of a large number of fragments or small number of nodes, it is also possible that some of the fragments are left without being replicated because of the T-coloring. As discussed previously, T-coloring prohibits storing the fragment in neighborhood of a

node storing a fragment, resulting in the elimination of a number of nodes to be used for storage. In such a case, only for the remaining fragments, the nodes that are not holding any fragment are selected for storage randomly.

IV. Conclusion:

In the proposed methodology, a cloud hosting and 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.

REFERENCES

[1]. 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

[2]. **"**DROPS: Division and Replication of Data in Cloud for Optimal Performance and Security"

[3]. Quanlu Zhang, Shenglong Li, Zhenhua Liy, Yuanjian Xingz, Zhi Yang, and Yafei Dai, **"**CHARM: A Cost-efficient Multi-cloud Data Hosting Scheme with High Availability".

[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

[5]. 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"

[6]. Z. Li, C. Jin, T. Xu, C. Wilson, Y. Liu, L. Cheng, Y. Liu, Y. Dai,and Z.-L. Zhang, "Towards Network-level Efficiency for Cloud Storage Services."

[7]. A. Bessani, M. Correia, B. Quaresma, F. Andr´e, and P. Sousa, "DepSky: Dependable and Secure Storage in a Cloud-of-Clouds."