DROPS Division and Replication of Data in Cloud for Optimal Performance and Security

ABSTRACT

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 paper, we propose division and replication of data in the cloud for optimal performance and security (DROPS) that collectively approaches the security and performance issues. In the DROPS methodology, we divide a file into fragments, and replicate 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. Furthermore, the DROPS methodology does not rely on the traditional cryptographic techniques for the data security; thereby relieving the system of computationally expensive methodologies. We show that the probability to locate and compromise all of the nodes storing the fragments of a single file is extremely low. We also compare the performance of the DROPS methodology with 10 other schemes. The higher level of security with slight performance overhead was observed.

**EXISTING SYSTEM**

* The data migration to the cloud is performed by the Iris file system. A gateway application is designed and employed in the organization that ensures the integrity and freshness of the data using a Merkle tree. The file blocks, MAC codes, and version numbers are stored at various levels of the tree. The proposed technique in [10] heavily depends on the user0s employed scheme for data confidentiality. Moreover, the probable amount of loss in case of data tempering as a result of intrusion or access by other VMs cannot be decreased.
* Our proposed strategy does not depend on the traditional cryptographic techniques for data security. Moreover, the DROPS methodology does not store the whole file on a single node to avoid compromise of all of the data in case of successful attack on the node.
* The authors in [11] approached the virtualized and multi-tenancy related issues in the cloud storage by utilizing the consolidated storage and native access control. The Dike authorization architecture is proposed that combines the native access control and the tenant name space isolation. The proposed system is designed and works for object based file systems. However, the leakage of critical information in case of improper sanitization and malicious VM is not handled. The DROPS methodology handles the leakage of critical information by fragmenting data file and using multiple nodes to store a single file.
* The use of a trusted third party for providing security services in the cloud is advocated in [22]. The authors used the public key infrastructure (PKI) to enhance the level of trust in the authentication, integrity, and confidentiality of data and the communication between the involved parties. The keys are generated and managed by the certification authorities. At the user level, the use of temper proof devices, such as smart cards was proposed for the storage of the keys. Similarly, Tang et al. have utilized the public key cryptography and trusted third party for providing data security in cloud environments [20]. However, the authors in [20] have not used the PKI infrastructure to reduce the overheads.
* The trusted third party is responsible for the generation and management of public/private keys. The trusted third party may be a single server or multiple servers. The symmetric keys are protected by combining the public key cryptography and the (k, n) threshold secret sharing schemes. Nevertheless, such schemes do not protect the data files against tempering and loss due to issues arising from virtualization and multi-tenancy.

**Disadvantages**

* + There is no Data Fragmentations to keep data in secure way.
  + The data outsourced to a public cloud is not secured due to lack of Cloud Security.

**PROPOSED SYSTEM**

* In the proposed system, the system collectively approaches the issue of security and performance as a secure data replication problem. The system presents Division and Replication of Data in the cloud for Optimal Performance and Security (DROPS) that judicially fragments user files into pieces and replicates 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 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. To further improve the retrieval time, we judicially replicate fragments over the nodes that generate the highest read/ write requests. The selection of the nodes is performed in two phases. In the first phase, the nodes are selected for the initial placement of the fragments based on the centrality measures. In the second phase, the nodes are selected for replication. The working of the DROPS methodology is shown as a high-level work flow in this system.

**Advantages**

* A successful attack on a node might put the data confidentiality or integrity, or both at risk.
* The system 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 a successful attack on the node leaks no significant information.

Methodology

In the DROPS methodology, user sends the data file to cloud. The cloud manager system (a user facing server in the cloud that entertains user’s requests) upon receiving the file performs:

(a) Fragmentation,

(b) First cycle of nodes selection and stores one fragment over each of the selected node, and

(c) Second cycle of nodes selection for fragments replication. The cloud manager keeps record of the fragment placement and is assumed

**SYSTEM REQUIREMENTS**

➢ **H/W System Configuration:-**

➢ Processor - Pentium –IV

➢ RAM - 4 GB (min)

➢ Hard Disk - 20 GB

➢ Key Board - Standard Windows Keyboard

➢ Mouse - Two or Three Button Mouse

➢ Monitor - SVGA

**Software Requirements:**

* Operating System - Windows XP
* Coding Language - Java/J2EE(JSP,Servlet)
* Front End - J2EE
* Back End - MySQL