An Efficient Cloud Security System Using Verifiable Decryption Process

F. Nivetha, PG Scholar
Department of Computer Science and Engineering
K.Ramakrishnan College of Technology, Samayapuram
Trichirappalli, India
kannan.nive5@gmail.com

P. Matheswaran, Assistant Professor
Department of Computer Science and Engineering
K.Ramakrishnan College of Technology, Samayapuram
Trichirappalli, India
mathesh3@gmail.com

***Abstract* - Now-a-days, the term cloud computing has the feel of a buzzword any more than the term the web is. Cloud computing is an emerging technology which provides a lot of opportunities for online distribution of resources or services. The most effective benefit of using cloud computing is higher availability of services with lower cost and easy scalability. In cloud computing, usually we transfer the data or submit the data to a third party administrator. It gives rise to security concerns. Even though, there are many security measures applied to protect data, still we are facing some security issues. Therefore, this paper proposes Division and Replication of Data in the cloud for Optimal Performance and Security (DROPS) that provide solution to the existing issues. In this methodology, the files in the cloud storage are encrypted and then divided into number of fragments and replicate the fragmented data over the cloud nodes. To enhance the security, cryptographic technique is used for encryption of data and index server is used to maintain the index terms of the fragments. So that no one can predict either the index terms or the encrypted fragment. Thus the attacker cannot get any meaningful information even in case of successful attack. Finally, it results with higher level of security with slight performance overhead.**

***Keywords* -- Cloud data, data fragmentation, node allocation, security, index terms**

I. INTRODUCTION

The cloud computing is a flexible, cost-effective and proven delivery platform for providing business or consumer IT services over the Internet. It combines a number of computing concepts and technologies such as Service oriented Architecture (SOA) [8]. It contains potential privacy risks which lead to privacy and security concerns because data is travelling over the internet and is stored in remote locations. Security is one of the most crucial aspects among those prohibiting the wide-spread adoption of cloud computing because essential services are often outsourced to a third-party, which makes it harder to maintain data security and privacy, support data and service availability [10]. The fundamental elements of the cloud require security which depends and varies with respect to the deployment models that is used, the way by which it is delivered and the character it exhibits [7].

The offsite data storage cloud utility requires users to move data in cloud's virtualized and shared environment that may result in various security concerns. Currently the security has lot of loose ends which scares away a lot of potential users. The security modules should cater to the entire issues element in the cloud should be analyzed at the macro and micro level and an integrated solution must be designed and deployed in the cloud to attract and enthrall the potential customers. Though, there are extreme advantages in using a cloud-based system, there are yet many practical problems which have to be solved [5].

The cloud manager will manage all the files that are all stored by multiple data owners. Once the file is stored in cloud, cloud manager will encrypt the file and then start fragmentation with the help of fragment engine. The fragmented files will be stored in cloud nodes using allocation techniques. The main advantage is that the data security gets increased because of storing each fragment in distinct location. Even a successful attack on a single node must not reveal the locations of other fragments within the cloud [10]. The amount of data compromise can also be reduced by making fragments of data file and storing them on separate nodes. The focus of this concept is on the security of the data in the cloud and we doo no take into account the security of the authentication system.

II. RELATED WORK

A. Juels et.al., [3] explained the gateway that manages cryptographic keys, maintains trusted storage for integrity and freshness enforcement, and may add redundancy to data for enhanced availability. Data integrity ensures that data retrieved by a tenant is authentic, that it hasn't been modified by an unauthorized party. These cryptographic protocols and auditing framework will provide security to the data and framework will verify an entire data collection without retrieving it from the cloud.

J.J. Wylie et.al. [10] tackled the difficult problem of reasoning about the engineering trade-offs inherent in data distributed scheme selection. In this p-m-n threshold scheme, data is encoded into n shares such that any m of shares can reconstruct the data and less than p reveal no information about the encoded data, where p be the encoded data , m be the original data, n be the n-way replication. This scheme is used to provide secret sharing of the data.

A. Mei et.al., [8] proposed a distributed algorithm for file allocation that guarantees high assurance, availability, and scalability in a large distributed file system. The algorithm can use replication and fragmentation scheme to allocate the files over multiple servers. In a fragmentation scheme, a file, f is split into n fragments; all fragments are signed and distributed to n remote servers, one fragment per server. This distributed allocation algorithm which will allocate the fragments to different remote servers and replicas will be maintained in same servers.

W.K. Hale [1] presented a minimum-order approach to frequency assignment is proposed traditional concept. Here, the assignment problems is modeled as both frequency-distance constrained and frequency constrained optimization problems. A frequency assignment is a function which assigns to each member of a set of transmitter an operating frequency from a set of available frequencies. The minimum-order approach and frequently assignment function are used to assign the nodes with minimum distance.

S.U. Khan et.al., [5] compared and analyzed heuristics to solve the fine-grained data replication problem over the Internet. The fine grained algorithms are used to determine where and how many replicas to be placed, so as to maximize the system performance. The decision where to place the replicated data must trade off the cost of accessing the data, which is reduced by having additional copies, against the cost of storing and updating the copies. The file allocation problem is for a network of M sites each with different storage capacity, replicate N files such that it satisfies the storage constraint and also optimizes some performance parameters.

III. SECURED DROPS CONCEPT

In the existing cloud system, the data are outsourced to a third-party administrative control, gives rise to security concerns. Due to attacks by other users and nodes within the cloud, the data compromise may occur [6]. In traditional cloud storage, the data owner sends copies of files over internet to the data serves, which the records the information to an off-site storage system that is maintained by a third-party. Whenever the data owner wants to retrieve the information, they access the data server through web-based interfaces. The concerns that are facing in existing system are reliability and security. To secure data, most systems use a combination of techniques, including: (i) Encryption is a complex algorithm is used to encode information, (ii) Authentication, which requires creating a user name and password, (iii) Authorization, the data owner lists the people who are authorized to access information stored on the cloud storage system. Even with these protective measures in cloud system, there's always the possibility that a hacker will find an electronic back door and access data. In order to overcome the problem with the existing system, the method called DROPS is used to the cloud storage system that contains node allocation algorithm that is designed to secure the data.

DROPS, a methodology that divides the owner's file into number of fragments and stores them in different locations, so that the attacker cannot tract the stored data. Even, the data was tracked means no meaningful information can be found. This methodology uses T-coloring algorithm, which aims at achieving high security for the data in the cloud storage. This will provide absolute value of the difference between two colors of adjacent vertices must not belong to fixed set of values.

Based on this algorithm, DROPS methodology can handle the attacks in which attacker gets hold of user data by avoiding or disrupting security defenses. The attacks that are handle by this methodology are data recovery, cross VM attack, improper media sanitization, E-discovery, VM escape, VM rollback. It is noteworthy that in case of successful attacks, no useful information can be revealed to the attacker because the attacker cannot trace the location of the fragment or the attacker cannot do anything with the information what is found [3]. The attacker can only keep on guessing the location of other fragments.

To enhance this security, a index server is maintained to store the index terms of all fragments and encrypted fragmented will be maintained separated. Both will get connected through heart beat protocol.

IV. SECURED DROPS FRAMEWORK

Figure 1.1 describes the system architecture. The client node is considered data owner which contains number of files. The data owner have to create an account in cloud system, so that the user can store their files in the cloud when there is internet connection exist.

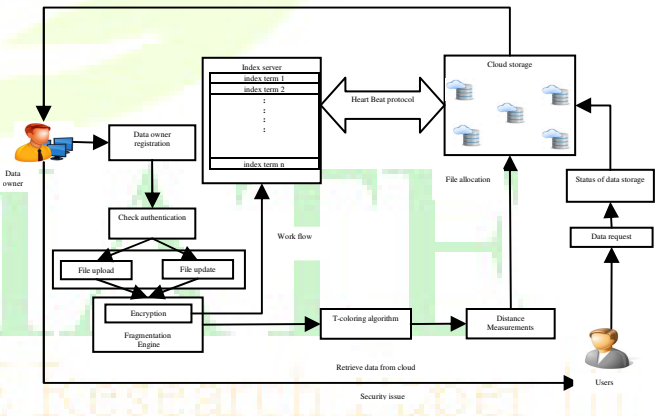

Figure 1.1 System Architecture

The cloud manager will manage all the files that are all stored by multiple data owners. Once the file is stored in cloud, the file will get encrypted. Then, cloud manager will start fragmentation with the help of fragmentation engine. Based on the fragmentation threshold value, the file will get

fragmented into number of pieces. Then it will be stored in cloud nodes using allocation techniques. Before that each fragment will get assigned with index terms. Each and every index terms will be maintained by a separate server named index server. After this, the primary node will be determined and it gets stored initially based on the index term. Then, all the remaining k^{th} fragments will be placed in remaining available nodes. The replication of the fragment will be maintained in same node. During the retrieval process, cloud manager will collect all the files and return to the requested user based on the index terms that is maintained.

Figure 1.2 defines the overall workflow of secured drops framework. The following steps will demonstrate the working of T-coloring algorithm:

(i) Once the data owner registered in the cloud storage, authentication has to be checked. The process has to be identified that the file has to upload or have to update.

(ii) Then, fragmentation engine is used to fragment the files into number of pieces. Using the centrality measures the index term of the primary node will be maintained in the index server.

(iii) Further using the T-coloring algorithm, the available nodes have to be collected as such the nodes that are not adjacent to the primary node are considered as available nodes.

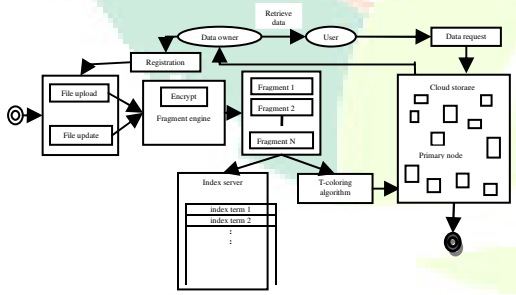

Figure 1.2 Workflow execution

(iv) The other nodes are considered as unavailable nodes, the remaining fragments should be placed in the available nodes based on the routine of T-coloring algorithm.

(v) Once all the fragments are placed in the cloud storage means its status have to be maintained by the cloud manager and the data owner also aware of the security status of the storage.

(vi) Here, the index terms of all the fragments will be maintained in separate server and the encrypted form of fragments will be maintained in cloud storage.

(vii) These index server and cloud storage will be located in different datacenters.

(viii) If any user request for data to the cloud storage means the data owner will get intimation about the request.

(ix) The data owner will provide a key to the user, so that the user can retrieve the file from cloud.

V. SYSTEM DESCRIPTION

The various components used here are: (A) Fragmentation Engine, (B) T-coloring algorithm, (C) Nearest Neighbor Node Identification, (D) Cloud management system. The working of each of the components is elaborated in the following section

A. Fragmentation engine

The fragmentation engine is an initial component which is located in the cloud storage. This component is mainly used for fragmenting the user files. Initially, data owner will upload a file to cloud storage system. The cloud manager will collect a file and encrypt the file using cryptographic techniques. Convert that encrypted file into 'n' number of fragments using fragments engine. All the fragments can upload to distinct node in different region. Once the file is fragmented, index terms will be provided for each and every fragment of file. Then, the primary node among the fragments should be determined. Based on the index term of primary node, the retrieval process can be done. Suppose the primary node was traced by intruder means nothing can be determined because without knowing the index terms, the node cannot be predicted. If an attacker is uncertain about the locations of the fragments, the probability of finding fragments on all of the nodes is very low. In cloud systems with thousands of nodes, the probability for an attacker to obtain a considerable amount of data reduces significantly. To improve the data retrieval time, fragments can be replicated in a manner that reduces retrieval time to an extent that does not increase the aforesaid probability.

B. T-coloring algorithm

The node identifier component uses the T-coloring concept for determining the available and unavailable nodes for allocation of fragments. Once the file is fragmented into number of pieces, the primary node is determined. Then, a non-negative random number will get generate and build the set T starting from zero to the generated random number. After that, the primary fragment should be placed in the cloud storage, based on centrality measures. Among the cloud nodes that are all available in the storage area, the central node should be identified, based on the betweenness centrality

measures. The set T is used to restrict the node selection to the nodes within the neighborhood at a distance belonging to T are assigned close_color and remaining nodes are assigned open_color.

By these notifications, the close_color nodes are considered as unavailable nodes and the open_color nodes are considered as available nodes. Initially, all of the nodes are given the open_color. Once a fragment is placed on the node, all of the nodes within the neighborhood at a distance belonging to T are assigned close_color. In the aforesaid process, we lose some of the central nodes that may increase the retrieval time but a higher security level can be achieved. If somehow the intruder compromises a node and obtains a fragment, then location of the other fragments cannot be determined.

C. Nearest-Neighborhood Node Identification

After determining the available nodes and unavailable nodes, the 'n' number of file fragments should be placed in a distinct node. Each and every fragment should have a size. Every single storage node calculate read and write a fragment, same time primary node stores primary copy of fragment. Here, every storage area has two field records, first field is to store primary node for primary fragment using centrality measures, and second field is to identify the nearest neighbor node for storing k^{th} fragment data using T-coloring and centrality measures.

The most central node to the cloud network should be choosing to provide better access time. The concept of centrality is used to reduce access time. Based on the available nodes that are determined and using the centrality values, remaining nodes can be placed. Once the nearest nodes are identified, the nodes should be verified that whether it is open_color node or close_color node. If it is open_color nodes means the k^{th} fragment can be placed, otherwise the nearest node to the close_color node should be identified and aforesaid process have to be followed, until all the fragments get placed in storage area.

D. Cloud management system

Once all the fragments are placed on its appropriate locations, the cloud manager should maintain all the nodes in the cloud. Cloud storage has a different unique storage in different region. This all node should be followed by a single primary node that represent first placement of fragment using its index term. Then, T-coloring algorithm is used to plan the remaining nodes and also it uses centrality measures. 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.

Based on replication algorithm, a controlled replication is performed to increase the data availability, reliability and improve data retrieval time. Cloud node store a fragment file in single time and replicate the file for single time because to reduce the storage cost for end user. Mainly, this cloud management system will maintain two different storage location or data centers. One is index server and other is cloud storage. Since, they are located differently no one can get any information about index terms or encrypted data.

VI. CONCLUSION

The cloud storage scheme collectively deals with the security and performance in terms of retrieval time. The user file was fragmented and the fragments are dispersed over multiple nodes. Once the fragment is placed in primary node, remaining nodes are placed over multiple nodes. The cloud manager will store and maintain that primary node for retrieval process. The fragmentation and dispersal ensured that no significant information was obtainable by an adversary in case of a successful attack. Each node in cloud should contain only one fragment. The nodes were separated by means of T-coloring. The performance of this system is in increasing manner to provide less access time. Mainly, this system will resulted in increased security level of data in cloud because of having separate index server and separate storage for encrypted fragment of data. Currently, with this security system, the user has to download the file, update the contents and again upload to the cloud. To overcome the aforesaid issues, as a future work, an automatic update mechanism that can identify and update the required fragment only. By this, the time and resource utilization will be saved.

REFERENCES

[1] Hale W.K. (1995), 'Frequency assignment: Theory and applications', *Proceedings of the IEEE,* Vol. 68, No. 12, pp. 1497-1514.

[2] Hashizume K., Rosado D.G., Fernndez-Medina E., and Fernandez E.B. (2013), 'An analysis of security issues for cloud computing', *Journal of Internet Services and Applications*, Vol. 4, No. 1, pp. 1-13.

[3] Juels A. and Opera A. (2013), 'New approaches to security and availability for cloud data', *Communications of the ACM*, Vol.56, No. 2, pp. 64-73.

[4] Kamara S. and Lauter K. (2010), 'Cryptographic cloud storage', *In Workshops on Real-Life Cryptographic Protocol and Standardization*.

[5] Khan S.U., and Ahmad I. (2008), 'Comparison and analysis of ten static heuristics-based Internet data replication techniques', *Journal of Parallel and Distributed Computing*, Vol. 68, No. 2, pp. 113-136.

[6] Loukopoulos T. and Ahmad I. (2004), 'Static and adaptive distributed data replication using genetic algorithms', *Journal of Parallel and Distributed Computing,* Vol. 64, No. 11, pp.1270-1285.

[7] Mazhar Ali, Kashif Bilal, Samee U.Khan, Bharadwaj Veeravalli, Keqin Li, and Albert Y. Zomaya (2015), 'DROPS: Division and Replication of Data in Cloud for Optimal Performance and Security', *IEEE Transactions on Cloud Computing*, No. 99, pp. 1.

[8] Mei A., Mancini L.V., and Jajodia S. (2003), 'Secure dynamic fragment and replica allocation in large-scale distributed file systems', *IEEE*

Transactions on Parallel and Distributed Systems, Vol. 14, No. 9, pp. 885-896.

[9] Subashini S., Kavitha V. (2011), 'A survey on Security issues in service delivery models of Cloud Computing', *J Netw Comput Appl* 34(1):1–11.

[10] Wylie J.J. , Bakkaloglu M., Pandurangan V., Bigrigg M.W., Oguz S., Tew K., Williams C., Ganger G.R., and Khosla P.K. (2001), 'Selecting the right data distribution scheme for a survivable storage system', *Carnegie Mellon University, Technical Report CMU-CS-01-120*.