Implementation of Division and Replication of Data in Cloud

P. D. Patni
M.E. (CSE) Student,
P.E.S. College of Engineering,
Aurangabad, Maharashtra, India

S. N. Kakarwal, PhD
Professor, CSE Department,
P.E.S. College of Engineering,
Aurangabad, Maharashtra, India

ABSTRACT
In cloud system the data is outsourced on the cloud, this may create security issues. In this paper we propose Division and Replication of Data in Cloud (DRDC) which can take care of security issues without compromising the performance. In this system, file uploaded by the client is first encrypted then divided into fragments. Then these fragments are replicated over the cloud nodes. Fragmentation and replication is carried out in such a way that each node contains only a single fragment. Thus if any one of the node is intruded by hacker, no significant information is revealed, and thus security is maintained. To further increase the security, nodes are separated by T-coloring graph method. Due to the T-coloring, the effort needed by an attacker to breach the security is increased multiple times. In addition to this, in this paper we compare this system (DRDC) with other methodologies.

Keywords

Cloud security, data fragmentation, replication, T-coloring.

1. INTRODUCTION

The cloud computing concept has revolutionized the usage and management of the information technology world. This concept has many applications in various businesses and organizations. It is also useful for individual users. Despite its usefulness, it has many security concerns. Security concern is the main obstacle preventing the wide spread adoption of cloud computing. This security issues may be due to-

- The core technology's implementation (VM escape, session riding etc.)Cloud characteristics (data recovery issues, internet protocol issues, etc.) shared environment of cloud results in many security concerns [1].

The security of components of cloud ensures cloud security. To ensure highest level of security, the weakest component of system must be secured, because any weak entity can put the whole cloud at risk. So the security mechanisms should be such that it will be difficult for an attacker to retrieve data even after successful intrusion in the cloud. In addition, data leakage must also be minimized.

For a large scale system, there is data replication strategy to deal with data retrieval time, data availability and reliability. But there are many numbers of nodes involved in replication strategy which increase the chances of attack [2].

Thus both security and performance are of paramount importance for clouds. In division and replication of data in cloud (DRDC), both performance and security issues are mitigated. This system first encrypts the file then fragments the file into pieces and replicates them and placed at distinct nodes within the cloud. There is no meaningful information in individual fragment increasing the data security. Further, the attack on a single node does not reveal the locations of other fragments. In addition to this, to ensure further security, the selected nodes should not be adjacent. T-coloring is used for separation of nodes. Replication of fragments over the nodes that generate highest read/write request improves data retrieval time [3].

Figure1: Division and replication of data in cloud (DRDC)

Figure 1 shows the working of division and replication of data in cloud (DRDC) and T-coloring. We are giving ten types of replication strategies as comparative techniques to this methodology (DRDC). These are-

A-star based replication strategies- i) DRPA-star, ii) weighted A star, iii) Aϵ star, iv) suboptimal A star, v) suboptimal A star2, vi) suboptimal A star3. Then bin packing based techniques like vii) local min-min, viii) global min-min. Then greedy based techniques like ix) greedy algorithm and x) genetic replication algorithm.

These techniques determine the number and locations of replicas and improve the performance. The data center network architecture mainly used in this study is three tiers.

In this paper we proposed a DRDC system which provides the security to the cloud and also increases the performance. This scheme first encrypts the file. Then fragmentation and replication takes place. Nodal placement of fragments is done with the help of T-coloring. This scheme ensures that no meaningful information revealed to the attacker even in case of successful intrusion of node. There is also control over replication of fragments, so that each of the fragments replicated only once so that performance is increased without decreasing security.

2. LITERATURE SURVEY

A technique presented by Juels et al. involves data migration to the cloud by iris file system. To maintain the integrity and freshness of data, a gateway application uses the merkle tree. In this technique the security of data mainly depend upon user's employed scheme. This system is unable to prevent data loss or access by other VM [4].

M. Tu et al. presented a secure and optimal placement of data objects in a distribution system. The encryption key is divided and division is done through threshold secret sharing scheme. This scheme pays attention to the replication problem with security and access time improvement. In this scheme data files are not fragmented and are handled as a single file. This scheme mainly focuses on encryption key security unlike our methodology [5].

G. Kappes et al. in their "Dike: Virtualization-aware Access Control for Multitenant File systems," presentation addresses virtualized and multitenancy related problems. This architecture works by combining the native access control and the tenant name space isolation. But in this system there might be leakage of critical information due to inadequate sanitization or improper VM. DRDC methodology involves fragmentation of data file and multi nodal storage of single file and thus prevents leakage of critical information [6].

D. Zissis et al. in their paper "Addressing cloud computing security issues," advises the use of a third party for security of cloud data. They use public key infrastructure, so that the level of trust is increased in the communication between the involved parties. Use of smart card was advised for the storage of the key at the user level. Tang et. al. also uses third party and public key cryptography for data security in cloud. But they do not use PKI infrastructure. The public key cryptography and the threshold secret sharing scheme are combined to protect the symmetric keys. However tampering and loss due to virtualization and multitenancy are not prevented [7].

Mazhar Ali et al. in their paper 'DROPS: Division and replication of data in cloud for optimal performance and security' advise fragmentation and replication of data and then use of T-coloring for multimodal placement of data. The nodes are selected on the basis of centrality measures that ensure an improved access time [8].

3. PROPOSED SYSTEM

If in a cloud, file is stored on a single node, there is significant risk for data security. In this kind of system the retrieval time can be decreased by replicating the files at multiple nodes. But this increases the risk of data security multiple times. Thus in order to increase performance in this system, we have to compromise the security.

To balance the security and performance DRDC methodology helps, as it does not store entire file on a single node. This system fragments the file and then replication of fragmented file is done. Thus even in a case of security breach, no significant information is revealed to an attacker. Replication is also in a controlled manner so that each fragmented file has only one replica, thus data security is not compromised in spite of maintaining performance [9].

There's a cloud manager in the DRDC system which is a secured entity. The cloud manager performs following functions –

- Receiving the file
- Encryption of file by AES algorithm
- Fragmentation of file
- Nodal selection and each fragment assigned a single node with the help of T-coloring.

Figure 2 shows the overall framework of DRDC method.

Fragments replication and storage of each replicated fragments over separate cycle of nodes again with the help of T-coloring. The user can decide the fragmentation threshold in terms of size or percentage. The user can divide the file efficiently so that no significant information is there in a single fragment. If the user does not give details of fragmentation threshold then default percentage threshold is set and used while fragmenting the file. The cloud manager has to pay attention to the communication channel between cloud manager and client, and make sure that the channel is secure.

As soon as file is divided into fragments, this system assigns the cloud nodes for each fragment. Centrality measures are employed to reduce the retrieval time. Three centrality methods are mainly used; these are betweenness, closeness and eccentric centrality. These measures may result in placement of fragments on adjacent nodes, thus compromises security. This is where the concept of T-coloring is justified. In this concept a set T is built starting from zero to random positive number. To make this system work, colors are given to the nodes. Let's consider there's an open_color before placing the fragment, as soon as fragment is placed on one of the node, then close_color is given to nodes surrounding the assigned node up to the T distance. This system makes cloud more secure, although somewhat performance is decreased due to less availability of central nodes.

Further in order to increase performance or to decrease the access time, data replication is done in a controlled manner. Again while placing the replicated fragments; concept of T-coloring is used. In data replication, some of the fragments may not be replicated due to T-coloring due to less number of nodes.

If client requests to download the uploaded file, then all the fragments are reassembled into a single file by cloud manager and then that file is sent to the client.

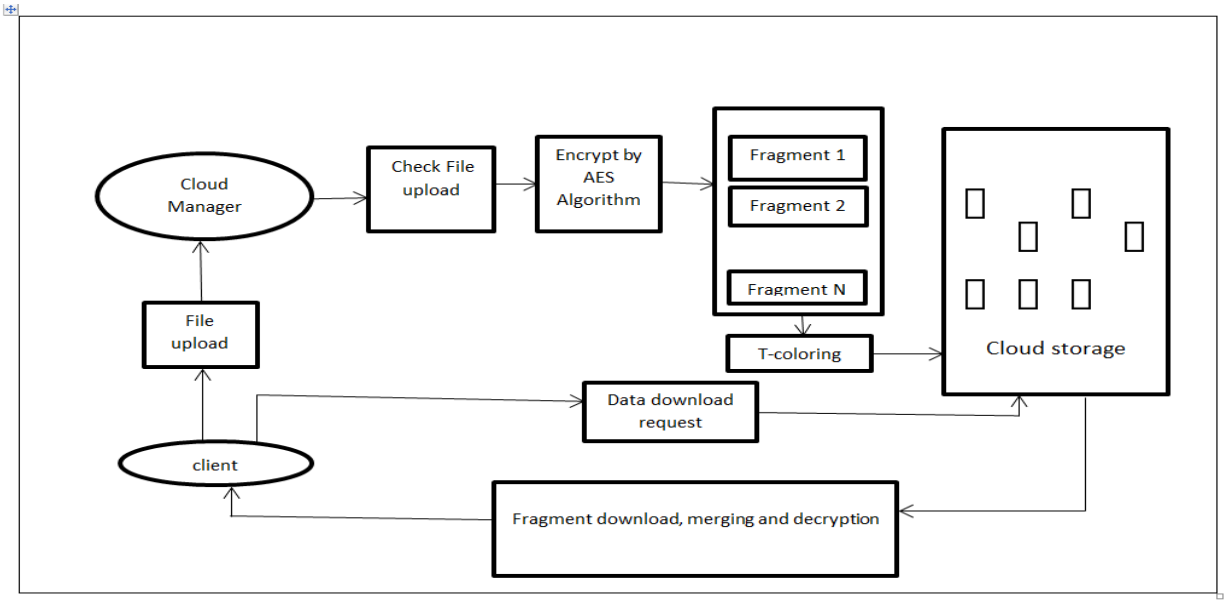

Figure 2: DRDC Framework

3.1 AES Algorithm

Advanced Encryption Standard (AES) is one of the most secure encryption algorithms available.

The features of AES are as follows −

-Symmetric key symmetric block cipher

-128-bit data, 128/192/256-bit keys

-Stronger and faster than Triple-DES

-Provide full specification and design details

- Software implementable in C and Java

ALGORITHM 1: Advance Encryption Standard [10].

1. Key Expansions—round keys are derived from the cipher key using Rijndael's key schedule. AES requires a separate 128-bit round key block for each round plus one more.

2. Initial Round

i) Add Round Key—each byte of the state is combined with a block of the round key using bitwise xor.

3. Rounds

i) Sub Bytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.

ii) Shift Rows—a transposition step where the last three rows of the state are shifted cyclically a certain number of steps.

iii) Mix Columns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.

iv) Add Round Key

4. Final Round (no Mix Columns)

i) Sub Bytes

ii) Shift Rows

iii) Add Round Key.

3.2 Fragmentation

Security of each node determines cloud security. An attack on one node makes easy for the intruder to attack on subsequent nodes in case of homogenous system.

But in heterogeneous system same effort will not result in intrusion of subsequent nodes. In this system attack on one node results in compromised security of information available only on that node, because in this system data file is fragmented and stored on different nodes. In addition to this, the possibility of finding fragments on all of the nodes is very less, if an attacker is not sure about fragment's location.

If number of nodes increases the probability of an intruder to obtain the data file decreases. If there are thousands of nodes in a cloud system, then that cloud system is relatively more secure.

3.3 T-coloring

In T-coloring graph vertices are colored. While coloring the vertices one thing is kept in mind that two adjacent vertices does not appear in one T field. For example, suppose 'a' and 'b' are two adjacent vertices, then coloring is done in such a way that 'a' and 'b' does not appear in T_{ab}, where T_{ab} is a set of nonnegative integers associated to the edge [a, b]. This is called vertex coloring.

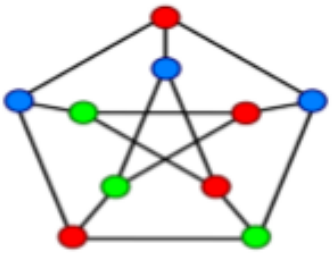

Figure 3. T-coloring node presentations.

Similarly, there's an edge coloring in which no two adjacent edges are of same color.

ALGORITHM 2: Fragment placement by T coloring [8]

Inputs and initialization**:**

$O = \{ O_1, O_2, \ldots\ldots, O_N \}$

$O = \{sizeof(O_1), sizeof(O_2), \ldots\ldots sizeof(O_N) \}$

$Col = \{open\_color, close\_color\}$

$Cen = \{cen_1, cen_2, \ldots\ldots, cen_M\}$

$Col \leftarrow open\_color \ \forall i$

$Cen \leftarrow cen_i \ \forall i$

Compute:

for each $O_k \in O$ *do*

Select $S^i | S^i \leftarrow indexof (max (cen_i))$

If $col_S^i = open\_color$ *and* $s_i >= o_k$ *then*

$S^i \leftarrow O_k$

$s_i \leftarrow s_i - o_k$

$col_S^i \leftarrow close\_color$

$S^{i\,\prime} \leftarrow distance(S^i, T)$

$Col_S^{i\,\prime} \leftarrow close\_color$

end if

end for

3.4 Replication

After fragmentation and nodal placement, fragments of file are replicated. Replication is in a controlled manner. So that only one replica of each fragment is made. Due to replication the access time is reduced hence performance is increased. Again nodal placement of replicated fragments is done with the help of T-coloring.

ALGORITHM3. For fragment's replication [8]

For each O_k *in* O ***do***

select S^i *that has max* $(R_i^k + W_k^i)$

if $col_S^i = open\_color$ *and* $s_i >= o_k$ *then*

$S^i \leftarrow O_k$

$s_i \leftarrow s_i - o_k$

$col_S^i \leftarrow close\_color$

$S^{i\,\prime} \leftarrow mdistance(S^i, T)$

$Col_S^{i\,\prime} \leftarrow close\_color$

End if

end for

3.5 System Model

i) Cloud client – client may be data owner or data user.

Data owner is a person who uploads file on cloud. Data owner knows fragment placement with their node numbers.

Data user is the person who downloads or view the files uploaded by others. Authentication is necessary for file accessing, otherwise he is considered as an intruder or attacker.

ii) Cloud server – When client uploads a file then cloud server performs many functions like encryption, fragmentation, nodal allocation, replication and further nodal allocation of replicas. When client requests a uploaded file for downloading then merging of fragments and decryption is done by cloud server.

Data Centre Network (DCN) used for communicational purpose in cloud. There are many DCN architectures mainly three tier, Dcell, fat tree etc. We mainly use Three tier architecture in this study [11].

4. COMPARATIVE TECHNIQUES

This methodology is compared with other methodologies like

i. DRPA-star – This is A-star based technique, for the data replication problem. It starts from the root, called the start node. Intermediate nodes provide the partial solution and leaf nodes provide the complete solution. Each node's communication cost is given by 'cost(n)= g(n)+h(n)' this cost for a node n is the estimated cost of the cheapest solution through n. Here g(n) is the search path cost from start node to the current node n and h(n), called the heuristic, is the lower bound estimate of the path cost from n to the solution. 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 called the OPEN list. This list is ordered in the ascending order so that the solution with the minimum cost is expanded first.

ii. WA-star – This is a refinement of DRPA-star which uses weighted function to estimate the cost. It is given as $f(n) = g(n) + h(n) + \epsilon [1 - (d(n)/D)] h(n)$, where d(n) is the depth of node n and D is the anticipated depth of the desired goal node. WA-star identifies solution within $1+ \epsilon$ of DRPA-star.

iii. Aϵ-star- This is also an extension of DRPA-star. The searching is focused on a particular space such that search would not deviate from optimal solution by a factor ϵ. This technique uses two lists OPEN and FOCAL. The FOCAL list is a sub-list of OPEN list and contains only those nodes that have f that do not deviate from lowest f by a factor greater than $1+ \epsilon$. The node selection is done from the FOCAL list. Aϵ star identifies a solution within a range of $1+ \epsilon$ of DRPA star.

iv. SA1 – (suboptimal assignment) this is DRPA-star based heuristic. In SA1, at level R or below, only the

best successors of node with the least expansion cost are selected.

v. SA2- This is also a DRPA based heuristic. In SA2, when the depth level R is reached for the very first time, best successor with the minimum cost are selected. Other successors are discarded.

vi. SA3- Again a DRPA based heuristic. Discarding done similar to SA2 except that the nodes are removed from the OPEN list except the one with the minimum cost.

vii. LMM (Local min-min) – Bin packing based technique. It sorts the file fragments based on the replication cost of the fragments. It assigns the fragments in the ascending order. If there's a tie, the fragment with lower size is chosen.

viii. GMM (Global min-min) – It chooses the fragment with global minimum of all the RC associated with the fragment. If minimum RC is same for two different fragments, then selection is done randomly.

ix. Greedy algorithm – Firstly algorithm searches through all of the cloud nodes. Then node with the minimum replication cost is selected for a file fragment. Then in second iteration this algorithm finds second node with lower RC, which in conjunction with the site already picked in first iteration.

x. GRA (Genetic replication algorithm) - This technique shows great stability in various scenarios. It mainly uses mix and match strategy. It consists of chromosomes representing various schemes for storing fragments over the nodes. In this technique selection, crossover and mutation operations are performed to select the best chromosome [12]

5. RESULTS AND ANALYSIS

If a single node stores all the information, then upon intrusion of that node there's a security risk of an entire data file. On the other hand if only a fragment of file is store on a single node, then only that fragment is revealed upon intrusion. In our methodology, person has to intrude large number of nodes to obtain significant data. This is because in our methodology fragments are stored on distinct nodes with the help of T-coloring. For an attack to be successful the number of nodes which are intruded must be greater than n.

There's an equation which determine an effort done by a person to attack a node.

$$E_{Total} = \min(E_{Auth}, n \times E_{Node}) \quad (1)$$

Where E_{Total} is effort necessary to breach the data; E_{Auth} is an effort needed to break in authentication and E_{Node} is effort needed to breach a single node.

In DRDC methodology we mainly pay attention to the data security in cloud, so we neglect the effort needed to breach authentication. So here we can conclude that an effort needed by a person to breach the security is directly proportional to the number of fragments [13].

To measure the performance of the system, we mainly relied on Response time (RT) which is total network transfer time. It depends upon two factors mainly- time due to read request and time due to write request. It is calculated by

$$RT = \sum_{i=1}^{M}\sum_{k=1}^{N}(R_k^i + W_k^i) \quad (2)$$

Where M is total number of nodes in the cloud, N is total number of file fragments to be placed, R_k^i is the total read request and W_k^i is total write request of k^{th} fragment of file.

We compared the results of DRDC system with the ten heuristic techniques. Study is done first by increasing number of nodes and secondly study is carried out by increasing number of fragments keeping the number of nodes constant. Values for ten heuristic techniques are directly taken from a paper mentioned in a reference.

We used windows azure platform for our study. Comparison is done by comparing RT values of our system with the ten heuristic data replication strategies. File size taken for our study is between 10kb to 5Mb.

5.1 Effect of increasing number of nodes

We studied the performance or RT by increasing the number of nodes. Numbers of nodes selected for simulation were 5, 25, 50, 100, 500 and 1024. RT values of our DRDC method and other ten heuristic techniques are given in table 1 against increasing number of nodes. Uploaded size of file taken for studying this effect is around 1Mb. From above results we conclude that our DRDC method has minimum response time and hence higher performance as compared to other technique as shown in figure 4.

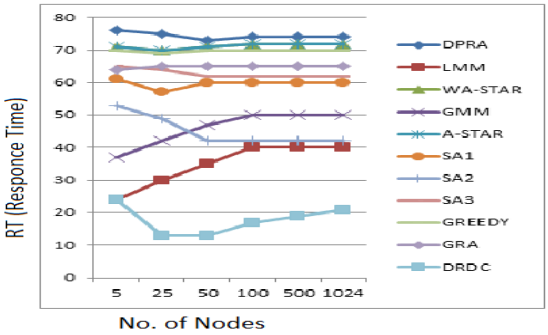

Figure 4: RT versus No of Nodes

5.2 Effect of increasing number of fragment

We studied the performance of these techniques by increasing number of fragments. Numbers of fragments selected were 10, 20, 40, 60, 80 and 100. Table 2 shows the RT values of the techniques with respect to increase in number of fragments. From this comparison we came to conclusion that at a given fragment number, the response time of our DRDC method is very less as compared to other heuristic techniques. Hence the performance of our DRDC methodology is more effective as shown in figure 5. Uploaded size of file taken for studying this effect is around 1Mb.

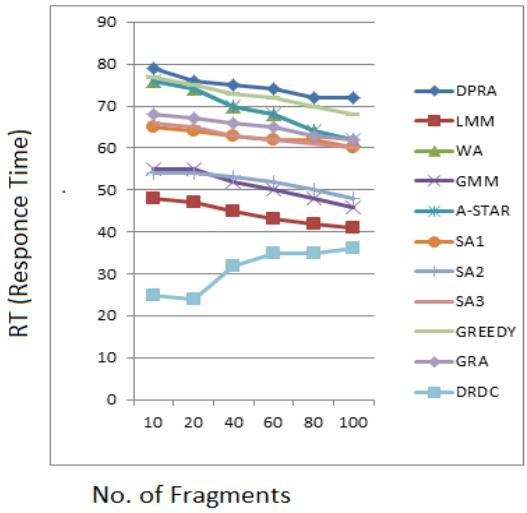

Figure 5: RT versus No. of file fragment

Table1. RT versus No of Nodes

No. of Nodes	Response Time (RT) %										
	DRPA	LMM	WA	GMM	AƐ	SA1	SA2	SA3	GREEDY	GRA	DRDC
5	70	24	71	37	71	61	53	65	70	64	24
25	75	30	70	42	70	57	49	64	69	65	13
50	73	35	71	47	71	60	42	62	70	65	13
100	74	40	72	50	72	60	42	62	70	65	17
500	74	40	72	50	72	60	42	62	70	65	19
1024	74	40	72	50	72	60	42	62	70	65	21

Table 2: RT versus No. of Fragments

No. of Fragments	Response Time (RT) %										
	DRPA	LMM	WA	GMM	AƐ	SA1	SA2	SA3	GREEDY	GRA	DRDC
10	79	48	76	55	76	65	54	66	77	68	25
20	76	47	74	55	74	64	54	65	75	67	24
40	75	45	70	52	70	63	53	63	73	66	33
60	74	43	68	50	68	62	52	62	72	65	35
80	72	42	64	48	64	62	50	61	70	63	35
100	72	41	62	46	62	60	48	60	68	62	36

7. CONCLUSION

In this paper we proposed a secured system for storage of data in cloud that is also very good in performance. The file which is uploaded on the server first encrypted, and then fragmentation and replication of fragments takes place. Nodes are assigned to the fragments and replicas with the help of T-coloring. Fragments are placed over the nodes in such a way that no node contains more than one fragment. This system of fragmentation and T-coloring increases the effort of an attacker to intrude the system. Even in case of successful attack on a fragment, no significant information is revealed to an attacker. Lastly we compared the performance of our method with the ten heuristic replication strategies.

8. REFERENCES

[1] K. Hashizume, D. G. Rosado, E. Fernndez-Medina, and E. B. Fernandez, 2013, “An analysis of security issues for cloud computing,” Journal of Internet Services and

Applications, Vol. 4, No. 1, pp. 1-13.

[2] L.M. Kaufman, 2009, "Data security in the world of cloud computing," IEEE Security and Privacy, Vol. 7, No. 4, pp. 61-64.

[3] W. K. Hale, 1980, "Frequency assignment: Theory and applications," Proceedings of the IEEE, Vol. 68, No. 12, pp. 1497-1514.

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

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

[6] G. Kappes, A. Hatzieleftheriou, and S. V. Anastasiadis, 2013, "Dike: Virtualization-aware Access Control for Multitenant Filesystems," University of Ioannina, Greece, Technical Report No. DCS2013-1.

[7] Y. Tang, P. P. Lee, J. C. S. Lui, and R. Perlman, 2012, "Secure overlay cloud storage with access control and assured deletion," IEEE Transactions on Dependable and Secure Computing, Vol. 9, No. 6, (Nov. 2012), pp. 903-916.

[8] 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, 2015, "DROPS: Division and Replication of Data in Cloud for Optimal Performance and Security," Concurrency and Computation: Practice and Experience, Vol. 25, No. 12, pp. 1771-1783.

[9] A. Mei, L. V. Mancini, and S. Jajodia, 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.

[10] Joan Daemen, Vincent Rijmen, 2002, "The Design of Rijndael: AES – The Advanced Encryption Standard," Springer, ISBN 3-540-42580-2.

[11] 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, 2013, "Quantitative comparisons of the state of the art data center architectures," Concurrency and Computation: Practice and Experience, Vol. 25, No. 12, pp. 1771-1783.

[12] S. U. Khan, and I. Ahmad, 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.

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