International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 3 Issue: 8
ISSN: 2454-4248
80 – 84

Cost Effective Information Dispersal and Retrieval Framework for Cloud Storage

Jaspreet kaur
Computer Science Department
UIET,PU,Chandigarh,Punjab
jas.cse92@gmail.com

Makhan Singh
Computer Science Department
UIET,PU,Chandigarh,Punjab
singhmakhan@pu.ac.in

Sarbjeet Singh
Computer Science Department
UIET,PU ,Chandigarh,Punjab
sarbjeet@pu.ac.in

Abstract— Cloud data storage applications widely demand security of data with minimum cost. Various cloud computing security threats supposed to be addressed in Cloud data service include Data Access Controllability, Data Confidentiality, and Data Integrity. In this paper, we propose a cost effective Information Dispersal and Retrieval framework for Cloud storage. Our proposed framework is different from existing approaches of replication. In our approach, multiple datacenters are considered as virtual independent disks for storing redundant data encoded with erasure codes and hence the proposed framework enables to retrieve user file even when failure of certain number of Cloud services occur . Besides security related benefits of our approach, the application provides user the cost-availability pattern of datacenters and allows cost effective storage on Cloud within user's budget limit.

Keywords- *Information Dispersal , Erasure coding, Cloud data storage plan chart, Cloud computing , Integer Linear Problem.*

*****

I. INTRODUCTION

Cloud computing is a type of Internet-based computing model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources (e.g., computer networks, servers, storage, applications and services) which can be rapidly provisioned and released with minimal management effort. On Cloud computing platform, resources are provided to the customers as service it guarantees to the Cloud subscribers according to the Service Level Agreement (SLA).Failures in Cloud is yet a common scenario therefore high availability, adequate performance and self-contained fault tolerant backup system are some important factors to be considered.

Concept of Information Dispersal is used in order to achieve high availability, adequate performance and self-contained fault tolerant backup system. Information Dispersal is the process of splitting original data into various parts so that they are unrecognizable as they sit in storage arrays or traverse the network and Data can be reassembled at the receiving device. Our approach ensures that maximum availability of user file is assured by dispersing the data parts as well as redundant information on multiple datacenters rather than explicitly storing on single Cloud service provider. User sets his budget above certain minimum predefined limit and expects maximum availability of data as per budget.

We proposed a framework for heterogeneous Cloud system. In this algorithm we address various issues of dispersal of information on multiple datacenters along with optimising the cost of storage on Cloud. As the datacenters in heterogeneous Cloud system has different availability rates and storage cost, optimising algorithm is used to disperse information on various such datacenters so as to meet the data availability requirements of end user defined in SLA.

The rest of paper is organized as follows. The related work is discussed in section 2. System model is presented in section 3. Section 5 has description of Erasure codes and Reed solomon coding. The proposed algorithm i.e. Information Dispersal and Retrieval Framework is described in section 5. Finally, Conclusion and Future Scope is discussed in section 6.

II. RELATED WORK

Information Dispersal is the process of splitting original data into various unrecognizable parts dispersed in different network locations and Data can be reassembled at the receiving device. H. Xu *et al*. Presented approach for secure and reliable data storage on Cloud using reed Solomon codes. In this paper they also discussed how to calculate optimal number of checksum parts required to add redundant information to user data files[1]. M.O.Rabin presented an approach for efficient dispersal of information for security, load balancing, and fault tolerance [2]. Gomez *et al.* presented scalable Erasure Coding algorithm with low computational overhead cost and a minimal amount of communication[3]. Khan *et al.* guided for load balancing and incremental scalability in data centers by applying erasure coding techniques in Cloud systems [4].Though these approaches can significantly enhance Cloud data reliability but end user gets no support to deal with performance of service providers. Lot of work is done on securing Cloud data, to which our approach is closely related. Hwang and Li proposed to protect shared Cloud data objects with use of data coloring and software watermarking techniques [5]. Their approach provides sole access to user to their Cloud data and also prevents data from being damaged, stolen etc. Shue *et al.* presented a Cloud-based system which distributes workload uniformly across virtual machines to improve the system performance and maximize utilization [6]. Adi Shamir proposes another technique which enables robust key management schemes construction for cryptographic systems [7]. Hugo krawczyk presented m-threshold scheme, where *m* shares recover the secret but *m* - 1 shares give no (computational) information on the secret[8]. These schemes ensure security and reliability even when half of pieces are destroyed and all except one of the remaining pieces are exposed. Research has been done on secure and reliable Information dispersal but very little has been done considering the effect of storage cost paid by user for dispersing information on various datacenters.

 ISSN: 2454-4248
80 – 84

To demonstrate the effectiveness of our proposed approach, we represented a framework using three types of datacenters, which allows cost effective, secure and reliable data storage on Cloud.

III. SYSTEM MODEL

Our approach proposes a framework for heterogeneous Cloud system. As shown in Figure 1, datacenters in one tier will have different configuration then datacenters in another two tiers. In order to disperse information for maximum availability, all data pieces and parity pieces for user's file are distributed on multiple datacenters.

As shown in Figure 1User sends request to Broker. Broker interacts with Cloud Information System (CIS) to retrieve list of registered datacenters along with other necessary details of storage cost and availability factor of datacenters. Broker passes on user's file, user's budget and information from CIS to scheduler which further is used to sort out those datacenters with maximum availability factor and cost of storage under user's budget. Scheduler finally provides the list of datacenters to Broker on which user's data could be dispersed efficiently. Basic unit of storage is Block.

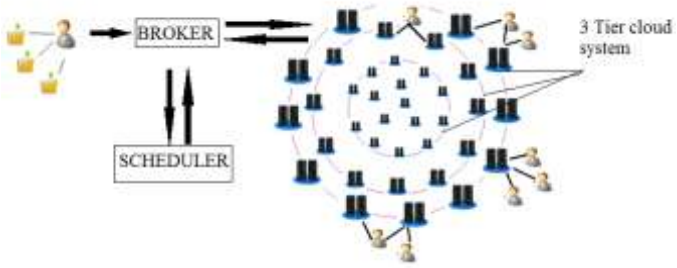

Figure 1. High level diagram for Information Dispersal Framework

Let there exist three types of datacenters as shown in three tiers in our framework and let $\{dc_1,dc_2,\ldots\ldots\ldots dc_i\}$ be a set of i datacenters ($i \geq 2$) in Cloud environment .Consider the fixed block size on all datacenters be 'b' , cost of storage per block in three tiers be $\{c_1,c_2,c_3\}$ and value $\{v_1,v_2,v_3\}$.Value factor of datacenters is basically used as ranking which is decided by service provider. Let $\{f_1,f_2,\ldots f_j\}$ be a set of j different files .Let $\{u_1,u_2,\ldots..u_r\}$ be a set of all users registered in Cloud environment. Suppose user u_k wants to upload file f_k of size S_k on Cloud with n_k data parts and m_k optimal parity parts which are calculated as discussed in section 4. Total number of parts of file f_k are (n_k+m_k).Cloud data storage plan chart is provided to user to disperse (n_k+m_k) parts on multiple datacenters. User chooses a suitable plan and submits budget B_k within cost range of selected plan. Broker invokes scheduler and passes user budget as well as opted plan to scheduler. Scheduler provides set of datacenters for information dispersal with total storage cost under user's budget B_k. Data files n_k+m_k are allotted to datacenters in set £ , discussed in section 5. As none of the datacenter has the complete knowledge about the user's data, this approach effectively defends security attacks from any single Cloud service provider.

On other hand, when user u_k wants to download the stored file f_k ,initially n data pieces of f_k are tried to be downloaded from all available datacenters in set £ on which file pieces of f_k were dispersed. In case if all n data pieces are available ,they could be efficiently combined to original file f_k without any decoding .Whereas in other way round, if one or more datacenter fails, then automatically all available data pieces n_k' and parity parts m_k' are downloaded. As long as $n_k'+m_k' \geq n_k$, the original file f_k could be retrieved by decoding missing data pieces using available n_k' data pieces .It's to be noted that parity pieces serve as redundant information and makes this framework reliable and fault tolerant.

IV. ERASURE CODES AND REEDSOLOMON CODING

A. Erasure code

Data replication was used as a fault tolerant technique in earlier days. Multiple copies of same data was stored on multiple servers to make storage more reliable. Data replication leads to ever increasing amount of Cloud data ,thus becoming impractical to implement. Erasure codes also known as forward error codes is a method of data protection in which using mathematical functions the data is fragmented, expanded and encoded with redundant data pieces and stored across a set of different locations on storage media [9].The purpose of erasure coding is to enable data that becomes corrupted at some point in the storage process to be reconstructed by using information about the data that's stored elsewhere in the vectors. Advantage of Erasure codes over traditional RAID is its ability to reduce the time and overhead required to reconstruct data.

B. Reed soloman coding

All n data parts of original file are encoded into m checksum pieces using Reed solomon algorithm such that out of all n+m pieces, any n pieces are enough to recover original n data parts [9]. Here n is n_k data parts and m is m_k checksum parts of user file f_k. To calculate checksum parts, firstly ((m+n) × n) vandermonde matrix A is created as in (1), where (i,j)th element in A is ij [9]. According to this definition, after deleting m rows of A, the left matrix is invertible. Secondly, ((m+n) × n) information dispersal matrix P is generated by applying finite sequence of row elementary operations on A. Note that rank of matrix is not changed on applying elementary matrix transformation and as each row in A is linear independent, thus information dispersal matrix P too maintains this property of matrix A such that on deleting m rows from P, newly formed matrix is invertible. The matrix P is as shown in (2), where I is an identity matrix and R is m×n matrix.

$$\mathbf{A} = \begin{bmatrix} 1 & \cdots & 0^{n-1} \\ \vdots & \ddots & \vdots \\ 1 & \cdots & (n-1)^{n-1} \\ --- & --- & --- \\ 1 & \ldots & n^{n-1} \\ \vdots & \ddots & \vdots \\ 1 & \ldots & (m+n-1)^{n-1} \end{bmatrix} \quad (1)$$

$$\mathbf{P} = \begin{bmatrix} 1 & \cdots & 0^{n-1} \\ \vdots & \ddots & \vdots \\ 1 & \cdots & (n-1)^{n-1} \\ --- & --- & --- \\ f_{0,0} & \ldots & f_{0,n-1} \\ \vdots & \ddots & \vdots \\ f_{m-1,0} & \ldots & f_{m-1,n-1} \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ -- \\ \mathbf{R} \end{bmatrix} \quad (2)$$

$\{d_0,d_1,\ldots\ldots\ldots d_n\}$ represent a vector of Data parts of length n and $\{c_0,c_1,\ldots\ldots\ldots c_m\}$ is a vector of Checksum parts of length m.

 ISSN: 2454-4248
80 – 84

With Information Dispersal matrix P , checksum vector C from data vector D is calculated as shown in (3.1) and (3.2), where $f_{i,j}$ for $0 \le i \le m-1$ and $0 \le j \le n-1$ are elements of m×n matrix F.

$$PD=\begin{bmatrix} I \\ -- \\ F \end{bmatrix} D = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \\ --- & --- & --- \\ f_{0,0} & \cdots & f_{0,n-1} \\ \vdots & \ddots & \vdots \\ f_{m-1,0} & \cdots & f_{m-1,n-1} \end{bmatrix} \begin{bmatrix} d_0 \\ \vdots \\ d_{n-1} \end{bmatrix} = \begin{bmatrix} d_0 \\ \vdots \\ d_{n-1} \\ -- \\ c_0 \\ \vdots \\ c_{m-1} \end{bmatrix}$$

$$= \begin{bmatrix} D \\ -- \\ C \end{bmatrix} = E \qquad (3.1)$$

$c_0=f_{0,0} * d_0 + f_{0,1} * d_1+.....+ f_{0,n-1} * d_{n-1}$
$c_1=f_{1,0} * d_0 + f_{1,1} * d_1+.....+ f_{1,n-1} * d_{n-1}$
....
$C_{m-1}=f_{m-1,0} * d_0 + f_{m-1,1} * d_1+.....+ f_{m-1,n-1} * d_{n-1}$ (3.2)

Suppose k parts are missing in vector D, where $k \le m$.On deleting k elements from D and m-k elements from C, new vector E' as in (4), (n-k) length vector D'= {d_0', d_1',..d'_{n-k-1}} and C' vector of length k is {c_0', c_1',... c'_{n-k-1}}.

$$E=\begin{bmatrix} \mathbf{D}' \\ -- \\ \mathbf{C}' \end{bmatrix} \qquad (4) \qquad P'=\begin{bmatrix} \mathbf{I}' \\ -- \\ \mathbf{R}' \end{bmatrix} \qquad (5)$$

$$P'D= \begin{bmatrix} \mathbf{I}' \\ -- \\ \mathbf{R}' \end{bmatrix} D = \begin{bmatrix} \mathbf{D}' \\ -- \\ \mathbf{C}' \end{bmatrix} = E' \qquad (6)$$

In equation (2), by deleting m rows from P corresponding to m rows deleted in E, another n× n matrix P' is derived as in equation (5), where I' is (n-k) × n matrix, and R' is k×n matrix. On deleting rows, equation (3.1) becomes equation (6). Inverse matrix L = P'-1 using Gaussian elimination method is calculated for invertible matrix P and data matrix D is recovered as in (7.1),(7.2) . $g_{i,j}$ are elements of n ×n matrix L. Where $0 \le i \le n-1$ and $0 \le j \le n-1$.

$$D=B'^{-1}E'=L\begin{bmatrix} \mathbf{D}' \\ -- \\ \mathbf{C}' \end{bmatrix} = \begin{bmatrix} g_{0,0} & \cdots & g_{0,n-1} \\ \vdots & \ddots & \vdots \\ g_{n-1,0} & \cdots & g_{n-1,n-1} \end{bmatrix} \begin{bmatrix} d_0 \\ \vdots \\ d_{n-k-1} \\ -- \\ c_0 \\ \vdots \\ c_{k-1} \end{bmatrix} \qquad (7.1)$$

$d_0=g_{0,0} * d'_0 + g_{0,1} * d'_1 +.....+ g_{0,n-1} * d'_{n-k-1} +$
$g_{0,n-k} * c'_0 + g_{0,n-k+1} * c'_1 +.....+ g_{0,n-1} * c_{k-1}$
$d_1= g_{0,0} * d'_0 + g_{1,1} * d'_2 +.....+ g_{1,n-k-1} * d'_{n-k-1} +$
$g_{1,n-k} * c'_0 + g_{1,n-k+1} * c'_1 +.....+ g_{1,n-1} * c'_{k-1}$
....
$d_n= g_{n-1,0} * d'_0 + g_{n-1,1} * d'_2 +.....+ g_{n-1,n-k-1} * d'_{n-k-1} +$
$g_{n-1,n-k} * c'_0 + g_{n-1,n-k+1} * c'_1 +.....+ g_{n-1,n-1} * c'_{k-1}$ (7.2)

After restoring vector D , vector C can be calculated using D vector and P vector as in equation (3.2)

V. COST EFFECTIVE INFORMATION DISPERSAL FRAMEWORK

Cost effective Information dispersal framework has 3 phases, which are discussed below:-

Phase 1:- **Calculating total number of data parts of user file**
Consider file f_k of user u_k of size S_k divided into n_k data parts ($n_K \ge 2$).Total number of data parts for file f_k is calculated as

$$n_k = \lceil S_k/b \rceil \qquad (8)$$

where b is fixed block size in all datacenters.

Calculating optimal number of checksum pieces

File parts ($n_k + m_k$) could be stored on N={1,2,3.....n_k+1} datacenters and the combination of N datacenters could be any of three types of datacenters available in system as per Table I. User is provided cost and value range for each scenario. To calculate storage cost and value, numbers of parity parts to be stored are required. Suppose m_k is number of parity parts and N datacenters in set £ = {dc_1,dc_2,…….dc_N} are allotted for storage of data .It's to be noted that n_k remains fixed for f_k and number of m_k checksum parts depends on N in each plan.

To retrieve original file, any n_k out of n_k+m_k parts are required and let X be the maximum number of datacenters which are allowed to fail or become unavailable simultaneously out of all N such that atleast n_k file parts of file f_k could be retrieved from available datacenters. Suppose ¥ is the failure set defined as follows:-

¥= P(£), where P(£) is power set of £ and ¥ ≤ X.The set of available datacenters € due to failure set ¥ can be defined as in equation (9)

$$€ = £ - ¥ \qquad (9)$$

To distribute n_k pieces over N datacenters in set £ where $2 \le N \le (n_k+1)$, number of data parts to be stored at £={d_1,d_2,…….d_N} respectively, are calculated as in equation (10).

$$ds^r = \begin{cases} \lceil n_k/N \rceil & when\ r = 1 \\ \left\lceil (n_k - \sum_{q=1}^{r-1} ds^r)/(N-r+1) \right\rceil & when\ 1 < r < N \\ n_k - \sum_{q=1}^{N-1} ds^r & when\ r = N \end{cases} \qquad (10)$$

when $n_k = ds_1 + ds_2 + ...ds_N$. equation (10) allows even distribution of n_k data parts over N datacenters such that $|nr-nq| \le 1$ for $1 \le r, q \le N$. Major requirement of fault tolerance is to ensure retrieval of original file from available datacenters in set € in case of failure of some X datacenters.

Let ps_1 , ps_2,....., ps_N are the number of checksum pieces dispersed at datacenters allotted £=d_1 , d_2 , ...d_N .For sure m_k = ps_1+ps_2+.....+ ps_N.To calculate minimum number of checksum pieces m_k, equation (11) containing Integer Linear Problem is to be solved.

$$\text{Minimize} \quad \sum_{i=1}^{N} ps^i$$
$$\text{Subject to} \quad for \text{ each failure set } ¥$$
$$\sum_{i \in €} ps^i \ge \sum_{j \in ¥} ds^j$$
$$\text{Where } ¥=P(£) \text{ and } |¥|= X. \qquad (11)$$

Note that solution to this optimal problem, automatically fullfills fault tolerance requirement when |¥|< X. For each distinct value of N, maximum number of file parts 'y' stored at any datacenter is calculated as follows

$$y= \left\lceil \frac{n_k+m_k}{N} \right\rceil \qquad (12)$$

 ISSN: 2454-4248
80 – 84

Phase 2: Preparation of the Cloud data storage plan chart

For storage of all file parts (n_k + m_k), a chart is prepared corresponding to allotted number of N datacenters on which information is dispersed. This chart displays cost range and value range corresponding to N where $2 \leq N \leq (n_k+1)$.The cost and value depends on type of datacenter chosen i.e ordinary datacenter, super datacenter, main datacenter. Cost range and value range for each distinct value of N is calculated using following algorithm where $2 \leq N \leq (n_k+1)$.

Input :- c_1,c_2,c_3,v_1,v_2,v_3
Output :- min, max, min_avail, max_avail
1. min=∞ ,max = -∞,z=0;
2. If(N≥2)
3. For i=N to 0
4. For j=N-i to 0
5. K=N-(i+j)
6. $C_N[z]=(i\times c_1 + j\times c_2 + k\times c_3)\times y$ and store i||j||k pattern for each iteration
7. $V_N[Z]= (i\times v_1 + j\times v_2 + k\times v_3)$
8. If($C_N[z]$ <min)
9. Min=$C_N[z]$
10.min_avail = $i\times v_1 + j\times v_2 + k\times v_3$
11.end if
12. If($C_N[z]$ >max)
13. max= $C_N[z]$
14. max_avail = $i\times v_1 + j\times v_2 + k\times v_3$
15. increment z , end if
16. end for
17. end for
18. end if
19. return (min, max, min_avail, max_avail)

Minimum and maximum cost as well as value are calculated out of all possibilities C_N^{N+2} of selecting N datacenters.Cost of storage for each possibility is stored in vector $C_N[]$ and value of each possibility is stored in vector $V_N[]$.Chart displays cost range and value range for each distinct value of N. As file parts are evenly distributed on allotted N datacenters using equation (10), hence overall total storage cost is calculated by multiplying storage cost per block for each datacenter with y as in (11).

TABLE I. DETAILS OF DATACENTERS

S No.	Datacenter category	number of datacenters	storage cost per block	value per datacenter
1	Super Datacenter	2	100	5
2	Main Datacenter	2	60	3
3	Ordinary Datacenter	2	30	2

TABLE II. CLOUD DATA STORAGE PLAN CHART

plan id.	No. of allotted datacenter	data parts distribution	parity parts distribution	cost range	value range
1	2	(2,2)	(2,2)	240-800	4-10
2	3	(1,1,2)	(1,1,0)	180-600	6-15
3	4	(1,1,1,1)	(1,1,0,0)	240-800	8-20
4	5	(0,1,1,1,1)	(1,1,1,0,0)	300-1000	10-25

As an example when user file is of size 240 and block size 60. Total number of data parts will be 4 and optimal parity parts corresponding to distinct value of N, cost and value are calculated as in Table II.

Phase 3: Distribution of data and checksum pieces over multiple datacenters

User u_k selects one suitable plan from Cloud storage plan chart prepared in phase 2 and submits budget B_k to broker within range of opted plan. Scheduler receives budget B_k, opted plan id from broker. Cost and value for all possibilities corresponding to N are already stored in vector $C_N[],V_N[]$.ordinary datacenters are closest to user and have highest value and they are cheapest as well. On other hand, Main datacenters are costly in comparison to Ordinary datacenter and possess least value. Super datacenters are costliest but still have highest value due to maximum availability. Thus optimal allotment amongst opted plan with minimum cost and maximum availability factor within budget B_k is calculated using following algorithm.

Input :- B_k
Output :- d_allot
1. temp = -∞;
2. For i=0 to length($C_N[]$) and $C_N[] \leq B_K$
3 If($V_N[i]$>temp)
4. temp= $V_N[i]$
5. d_allot = i||j||k pattern for datacenter allotment
6. end if
7. return (d_allot)

Scheduler recalls resultant pattern d_allot = i||j||k in same plan id. Accordingly i super datacenters, j main datacenters and k ordinary datacenters are alloted on which file parts (n_k+m_k) are stored. Where $C_N[]$ is vector storing cost of storage for all possibilities of N datacenters as per opted plan.$V_N[]$ is vector storing corresponding availability factor for all possibilities of N datacenters as per opted plan.

VI. CONCLUSION AND FUTURE WORK

In this paper, we addressed some major issues related to Cloud storage, including reliability, security and cost efficiency. Rather than using redundancy at server side, we presented cost effective and secure distributed Cloud storage framework for end users. In this approach redundant data files are calculated using erasure code techniques and uploaded on various network locations. The redundancy added to data files allows multiple failures of datacenters. We proposed an algorithm which solves cost efficiency issue. Space efficiency is achieved by forming optimal problem for calculating number of checksum pieces. As all parts of user's data are distributed on various datacenters, thus no single datacenter has access to complete user data. In this way, our approach protects data from unauthorized access in Cloud and security issue is resolved. Besides advantages of security and fault tolerance, our approach provides cost effective framework for Cloud data storage.

For Future work, other security techniques could be applied on Cloud data such as digital signature technique to ensure no data alterations at end of service providers. Furthermore, it would be feasible to optimise number of data pieces to be calculated for user file as per conditions of datacenters for information dispersal under suitable storage cost.

 ISSN: 2454-4248
80 – 84

REFERENCES

[1] H. Xu ,D. Bhalerao, "Reliable and secure Distributed Cloud data storage using Reed Solomon codes", International Journal of Software Engineering and Knowledge Engineering ,(2015).

[2] M.O.Rabin, "Efficient Dispersal of Information for Security,Load Balancing, and Fault Tolerance", Journal of the Association for Computing Machinery, April 1989.

[3] L. B. Gomez, B. Nicolae, N. Maruyama, F. Cappello and S. Matsuoka, "Scalable Reed-Solomon-based reliable local storage for HPC applications on IaaS Clouds", in Proc. of the 18th International Euro-Par Conference on Parallel Processing (Euro-Par '12), Rhodes, Greece, August 2012, pp. 313-324.

[4] O. Khan, R. Burns, J. Plank and W. Pierce, "Rethinking erasure codes for Cloud file systems: minimizing I/O for recovery and degraded reads", in Proc. of the 10th USENIX Conference on File and Storage Technologies (FAST-2012), San Jose, CA, USA, February 2012, pp. 20-33.

[5] K. Hwang and D. Li, "Trusted Cloud computing with secure resources and data coloring", IEEE Internet Computing 14 (5) (2010) 14-22.

[6] D. Shue, M. J. Freedman and A. Shaikh, "Performance isolation and fairness for multi-tenant Cloud storage", in Proc. of the 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI '12), Hollywood, CA, USA, October 8-10, 2012, pp. 349-362.

[7] A. Shamir, "How to share a secret", Communications of ACM, November 1979.

[8] Hugo Krawczyk, "Secret Sharing Made Short" , IBM T.J. Watson Research CenterYorktown Heights, NY 10598

[9] Haiping Xu ,Deepti Bhalerao, "Reliable and secure distributed cloud storage using reed Solomon codes" , International Journal of Software Engineering and Knowledge Engineering © World Scientific Publishing Company