I J C T A, 9(28) 2016, pp. 307-317

RABMYST–PERFORMANCE AND ANALYSIS OF RABIN'S MYSTIC SHARING ON MULTI-CLOUDS

R. Sugumar* A. Rajesh and R. Manivannan*****

Abstract: Cloud computing is the power that extent its support to many individuals and organization, where user data is available anywhere in the world, when user wants it, but users not trusted on Cloud Service Providers (CSPs), because of security concerns. As many researchers and organizations implement their own protocol to secure their outsourced data. Here we propose a method RABMYST (Rabin's Mystic Secret Sharing), an Information Dispersal Algorithm (IDA), splits the encrypted file in to shares and replicates and distributes to N number of cloud servers,. Where user retrieve the original file by 'm' pieces i.e., $m = N - 1$ with the threshold 't', where any minimal parts that replicated and distributed to CSPs are required to reconstruct the file. Our scheme implemented in openstack swift object storage and analyzed with different file sizes and also with different parameters like increasing files size, access structure and key pairing with CSPs, and we analyzed in multi-cloud environment interface kaavo-IMOd, jira compared with different sharing schemes, different file sizes and performance on multiple CSPs. Our method provides users with security benefits and easy access structure to CSPs.

Keywords: RABMYST - Rabin's Mystic Secret Sharing, IDA- Information Dispersal Algorithm, CSPs- Cloud Service Providers, TPA-Third Party Auditor.

1. INTRODUCTION

Cloud computing has a wide variety of industrial standards as industries meets many high concentration of risks, safety for the outsourced data's are increasing attention from industry, research and authorities many of the organizations approached towards safety management as they were involved in many monitoring tools and innovative approach. During the last decade economic losses occurs from variety of reasons like disaster loss, data loss from cloud service providers, for this organizations developing their own protocol for storing data in the cloud for the security consideration.

Our method proposed to design a system where users/dataowners upload a file using Shamir's secret sharing algorithm to split a secret of file in to many pieces and shares of files are distributed and replicated to $n - k + 1$ CSPs. To reconstruct the pieces this will require $n - k$ CSPs, our design is to upload the file share using Kaavo to many cloud servers we define our protocol file share construction algorithm, consider a file $F = a_1, a_2, .. a_n$ all the a_i were streamed as integer to disperse the file in to n independent vectors($V_1, V_2, .. V_n$). In share reconstruction algorithm a file k integers at a time with CSPs (CSP_1, CSP_2.. CSP_n) to reconstruct the original file.

Most of the cloud service providers like google drive, dropbox, box etc., proved the popularity of cloud storages. In April 2013, Amazon storing 2 trillion objects in Amazon S3, however cryptographic encryption method of storing data provides computational complexity and complex key management issue

* Research Scholar, Department of Computer Science and Engineering, SCSVMV University, Kanchipuram, Tamilnadu, India. **Email:** sugumar_prof@rediffmail.com

** Professor and Head, Department of Computer Science and Engineering, C. Abdul Hakeem College of Engineering and Technology, Melvisharam, Vellore District,Tamilnadu, India. **Email:** amrajesh73@gmail.com

*** Professor and Head, Department of computer science and Engineering, Stanley college of Engineering and Technology for women, Hyderabad. **Email:** drmanivannan@stanley.edu.in

at the CSPs. Due to computation cost of key at the single clouds makes more complexity, our method uses multi-cloud environment at the desired availability of cloud cost as the user is able to pay for the service our technique is capable to disperse and retrieve file with guaranteed security. In the past several multi-clouds are existing with the feature what we have for example iDataGuard provides confidentiality, but redundancy feature makes too overhead problem and also depsky provides confidentiality and redundancy feature but having too overhead in computational features. Our rest of the paper is organized as follows with interface a Kaavo-IMOd, a Multi-cloud management tool for distributing files to cloud servers, constructing and reconstructing the shares.

2. RELATED WORK

As the security is the main concern in the cloud computing, many researchers found many ideas on the secret sharing scheme. A CloudStash scheme[1] splits the file into many shares of secret and distribute those shares in to multi-clouds simultaneously, where threshold shares need to reconstruct the file, it shows that this scheme is faster for small files and longer for larger files not statistically worse. In an multi-cloud environment where as, many number of CSP providers like Dropbox, GoogleDrive or interface like Multcloud. Users will interact the CSPs through Application Server[2], a file is splitted in to chunks of size and sends to the destination CSPs randomize every chunk of data and stores in a repository after that using an Randomization key, users can able to download the file by this proof of privacy is obtained.

Seung-Hyungseo et.al.,[4] proposed the certificateless encryption scheme without pairing operations for storing their data in public cloud. In MCC-PKE scheme they start with the setup stage after setting their private key and public key after that SEM-key extract, user has to register their identity their results showed efficiency in implementation and securely share their sensitive data's in public clouds. MingqiangLi[10] describes the confidentiality of Information dispersal Algorithm, an IDA method encodes a file in to unrecognized pieces, so that we can reconstruct the file from 'm' pieces, this systematic study makes the confidentiality in a practical application.

Sian-Jheng Lin and Wei-Ho Chung[13] used the coding technique for converting a file into 'n' digital shadows, here the IDA method with fast Fermat number transform (FNT) is to improve the performance by making decoding algorithm takes O(nlogk) or O(klog^{2}k). Andrew Tytula[12] proposed Rabin's IDA, where the file uploads efficiently to many peers and only subset of the peers are required to rebuild the original content. Many security concerns was proved like SC-001 File Confidentiality, Unauthorized user cannot view the file, next concert is SC-002 Index File Security, file cannot be lost if more than one piece is lost, next concert is SC-003 Viruses, next SC-004 File Integrity at least k pieces are there it is having capable of rebuilding a file.

Rafael M. de. O. Libardi et.al.,[11] proposed the method MSSF, a multi cloud storage selection framework it contains some basic algorithms, ser of security rules and allow user to store the data when they need, MSSF had great impact on users automatic services and user friendly multi-cloud data dispersal and transparency to the user. Hugo Krawczyk[20], Secret Sharing made short, he presents a m-threshold 'm' share need to recover the secret, but m-1 gives no idea on secret, secret 'S' will be |S|/m, here it was implemented in natural way traditional sharing schemes, Cryptographic methods and information dispersal, while robust secret sharing and verifiable secret sharing deals with potentially corrupted share and also with corrupted dealers of the secret.

Openstack swift integration along with Nippon Telegraph and Telephone Corporation (NTT) developed world's first high-speed secret sharing engine called "Super High-Speed-Secret Sharing (SHSS)[28], this technology makes high speed for fragmentation and reconstruction openstack swift allows the files to quickly store and retrieve files. To improve performance NTT developed 64-bit processor, so it was

50 times faster than previous Secret Sharing scheme. Tatiana et.al.,[14] proposed multi provider cloud architecture uses the medical record to store in the cloud, this features secret sharing as important measure to distribute Health records and it fragments with higher redundancy and additional security and privacy protection, they discussed Shamir's secret sharing scheme and Rabin Information dispersal algorithm shows low computational overhead and they recommended Rabin's approach gives smaller shares.

Table 1.
Different Security Mechanisms Over Cloud Storages

Security Measures	*Goal*	*Location*	*Performance*
Client/Server Authentication	Prevent unauthorization	All nodes	Faster, sometimes insecure.
Network security	Avoids eavesdropping	All nodes	Faster, Computational overhead
Identity management	Usability and reduce management overhead	Data owner to CSPs	Faster, sometimes insecure
Attribute based encryption	Protect sensitive information of individuals	Data owners to CSPs	Faster, computational overhead
Digital signature	Unauthorized access	All nodes	Overhead in key management to all the nodes
Hash function cryptography	Protecting information to outsiders	Data owners	Faster on hash function
Secret sharing	Prevents the data when it is also spied	Data owners/ users	Faster, Availability, No much computation overhead

3. SYSTEM ARCHITECTURE

Our system architecture is organized as user configuration, Rabin's IDA, uploading the shares to CSPs, after these steps user downloads the original file by reconstruction process. At user configuration, system has to check with our configuration to upload the files on multi-cloud server, as a part of configuration user has to register with many cloud service provider and had an account. Our scheme is proposed to distribute a piece of information among all 'N' active sites/nodes so, that reconstruction is also processed from 'm' active nodes.

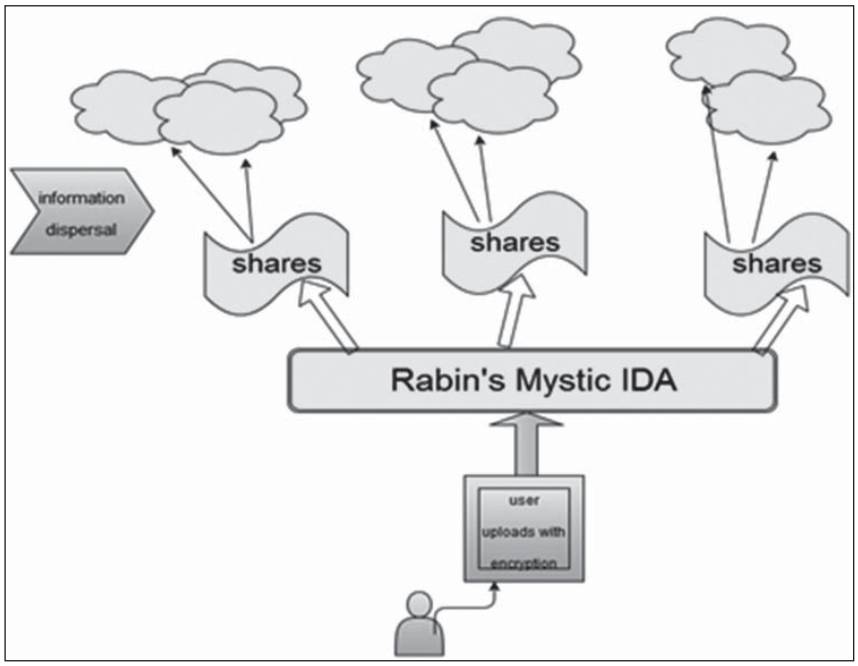

Figure 1. System architecture with Rabin's IDA secret sharing scheme

In this secret sharing scheme a secret S would be shared by 'N' number of shares (nodes) ie. Threshold value k, this scheme uses polynomial function as in the order (k-1). In the process of uploading a file, file is splitted in to different chunks of fixed size, after generate a randomization key and that will be stored in repository. Group the chunks as our system configured with number of CSPs and uploads to the cloud server. While downloading the file, at the threshold level of 'K' shares we may get the original file as we upload to CSPs. Users maintains the log on Multi-cloud interfaces to where the share gets replicated and distribute in the cloud servers. At the reconstruction process it collects from the minimal number of cloud server, where the shares are distributed in every cloud servers.

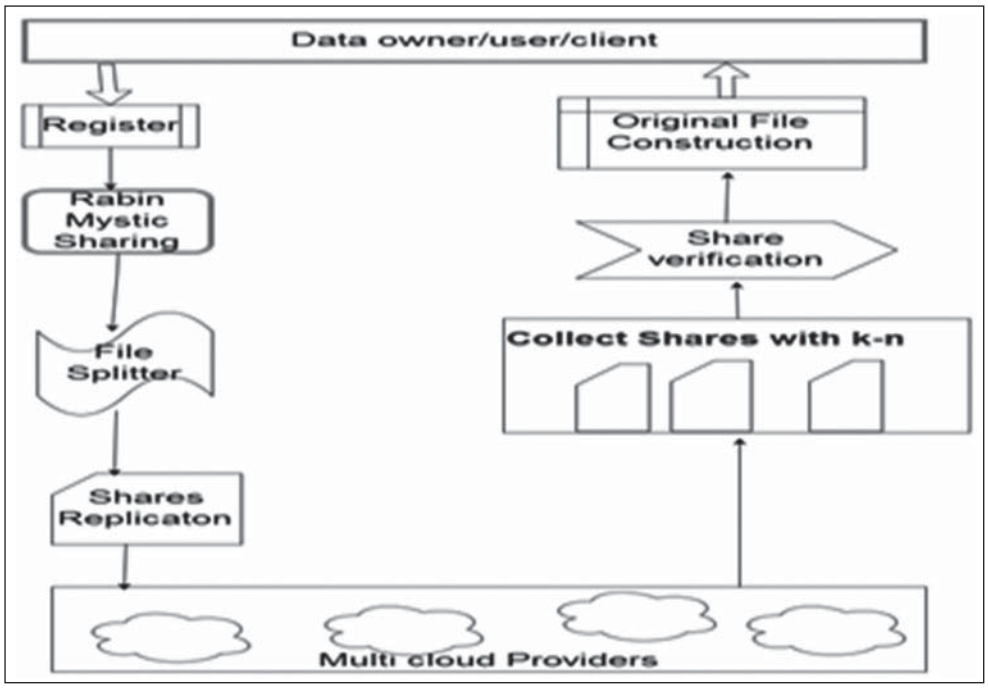

Figure 2. DFD for Rabin's mystic sharing scheme

Our system flow, a user/client/data owner has to register with mystic sharing scheme, as user wants to outsource the data, we conceive the Rabin's mystic sharing scheme with Kaavo-IMOd, a multi-cloud management tool interface and we splits the encrypted file with the regular shares and that will be uploaded to multiple CSPs. The shares thus stored on multi-cloud CSPs will replicate the shares to different CSPs, when it was shared that the same share will not be replicated on same CSPs, up to this user part of outsourcing the data is finished. After when user/data owner/client wants the data which was outsourced to CSPs, our scheme will collect the shares with (k-n) ie., from 'm' share and share will be verified with encrypted key with the users as it will efficiently reconstruct the original file.

Input: U, user encrypts the file with secret key S

Terms: R_i: service providers(CSP)
P_i: participant node in CSP.
r_i : replicate share.

Process:

splits secret key S in to N number of shares, $S_1, S_2 \ldots S_n$.
for each share $S_i \rightarrow S$, where i€1,2..n,
do
replicates $S_i \rightarrow k>=1$ (S_i, r_i)
distribute$S_i r_i \rightarrow$N no. of CSP_s

end for

for each CSP R_i, I€1…n.

do

selects pair of (R_i,S_i) from CSP

U, stores (S_i,R_i), when R_i holds S_i

while

check for (S_i,R_i) again not in same CSP (R_1,S_1)

end

Get unique solution m=X_i from (R_i, N_i)

Algorithm 1. File Splitter.

Our algorithm file splitter works on when an user outsource a encrypted file, our process split the key along with data into N number of shares, like S1,S2,…S*n*. Each share will have a unique identifier, so each share will gets replicated to different CSPs a and also our system gets checking on each share will replicate to different CSPs, not same CSP will have two or more copies of same file share. For each and every share replication and distribution to CSPs, we maintain the pair of share with CSPs, while user stores these pair for reconstruction process and we get a unique solution *m* = X*i* which will maintain all the pair of shares and CSPs.

In the reconstruction process when all the participant in the cloud clusters that will acts as CSP resource providers Ri, when user wants to download the file cloud service resource provider Ri query a pair of key from user, where user maintains the key for downloading a file with the unique solution which was maintained at file splitter using unique solution m= Xi, from all the cloud service provider this unites all the shares collectively and group them in to a single file, where original file was constructed from the dispersed environment.

for each CSP, Ri, i€1…n.

do

Ri query (I,Pi pair from user U and Ri has Ni

reconstruct from the unique solution m=Xi from (Ri, Ni)

unites shares Si from 'm' number of shares Si1,Si2,Si3…Sin.

endfor

Algorithm 2. File Reconstruction.

4. SYSTEM IMPLEMENTATION

Our implementation starts with the user registration with the CSPs, our system user/data owner registers with Kaavo-IMOd a Multi-cloud management tool, a interface which will provide many cloud servers instance which will have some amount of cloud storage in their servers. At the Rabin's IDA, data owner/ user gives a secret share to cloud service provider, user gives a share to each server and groups for some threshold 't' and with the only 'k' number of user's we can reconstruct the file by the polynomial function.

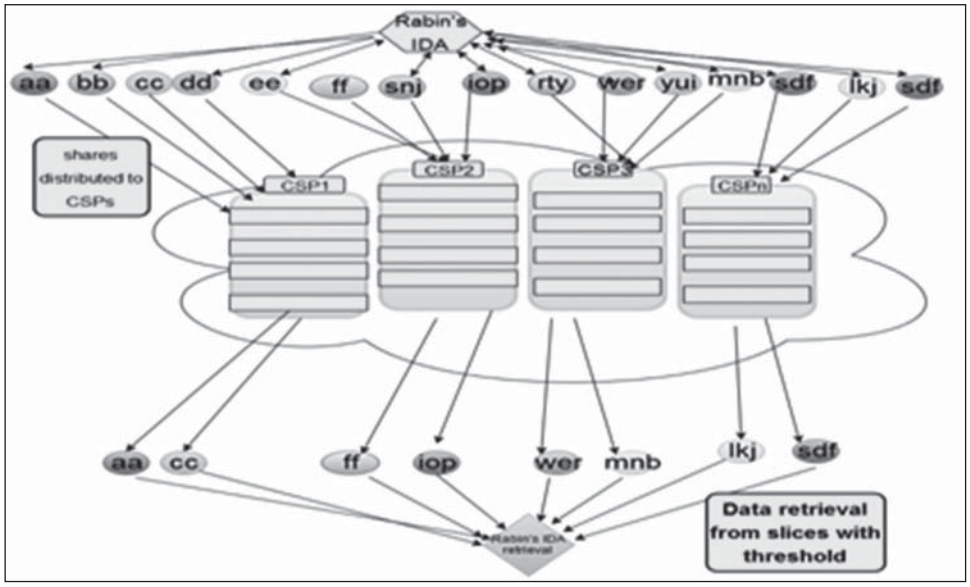

Figure 3. File Share distribution to CSPs

By using Rabin's Information dispersal we splits a encrypted file in to many number of pieces and distributed to N number of CSPs with the threshold 't' at the reconstruction process. User/Client can able to download the original file from the 'm' pieces of information at each CSPs.

$$f(x) = K_0 + K_1(x) + K_2(x^2) + \ldots d_{k-1}\, x_{k-1}$$

$$\Sigma^{K}_{i=0} = \{\pi_{i \neq n} X/X_j - X_i\} f(x)$$

Our scheme is divided in to pieces along with the secret shares, let we assume a file is divided in to four pieces and it was distributed among four cloud service providers, the diagrammatic illustration gives the impact how the reconstruction happens with the threshold '*t*' nodes

Figure 4. Mystic sharing among CSPs

Getting the pieces from any 2 nodes we can get all four pieces to reconstruct the file, when we maintain threshold '$t = 2$'. Our paper implementation is organized as per the algorithm discussed below.

Input: U, N, K, *t*, P = U*i* = 1*k* Xi
U = user
N = Number of nodes
K-secret shares of N-1 at threshold '*t*'
P-possible permissions from the CSPs

```
Data:    A file 'F'
Result:  N shares, stored in cloud servers.
Algorithm:
        for each file do
        Apply secret share, split into N shares where, i=1,2,…N .
    begin
        place each share Ni at interface to multi-cloud environment
        distribute the share to required number of CSPs
        replicates the share to CSPs
            checkSi ≠Ni(once)
            Si=Ni+1
    end
    for every share S=Si,Si+1..Sn
        set threshold 't'
        from N-S+1 collect all the shares
        construct Si,Si+1…Sn=file (original)
    end
end
```

Algorithm 3. Algorithm for Rabin's Mystic Sharing Scheme Implementation

5. EXPERIMENTAL EVALUATION

Our experimental setup constructed in lab with openstack environment swift object storage, the system configuration with processor intel core i5 CPU 240 GHz, 8GB RAM and 64-bit ubuntu operating system. First we analyzed the performance with the openstack consistent hashing, a ring it consists of space available at all possible computed hash values divided in equal partitioned size. In openstack object storage swift, we formed 4 partitions and computing has (object) modulo4 for storing the files, each partitions was considered as the CSP. In our cases we identified zones, an isolated place where likely to be a server, a disk or whole cabinet swift supports the encryption of object data at Rest on storage nodes. The encryption_root_secret option holds the master secret key used for encryption, a keymaster is authenticated to issue key to the users for storing the data in cloud storages. Our file splitter splits the encrypted key, keywrapping feature in openstack enable shares to wrap the key and shares. Our analysis states with the performance based on different size of pieces and access structure of pairing key with the nodes in clusters.

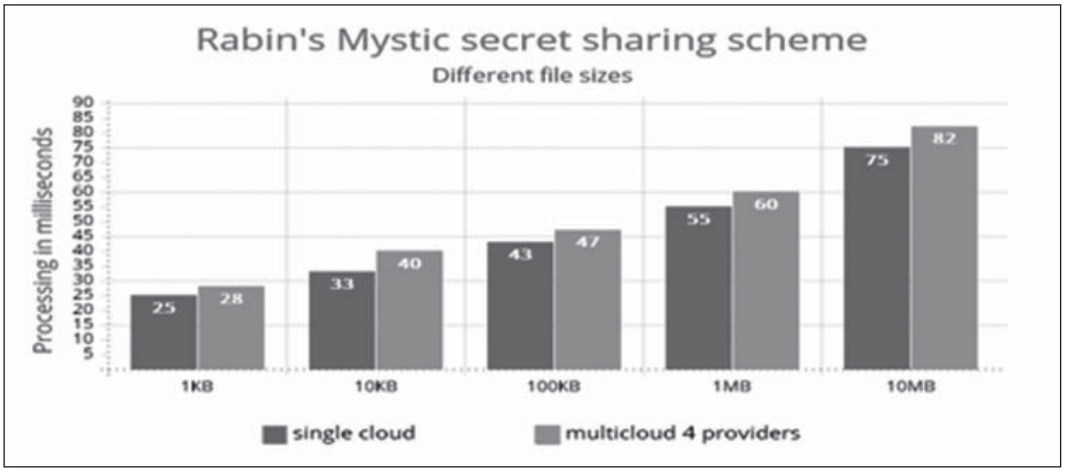

Figure 5. Rabin's Mystic Secret sharing scheme execution

Our analysis made at netstat and netcat a network monitoring tool, after analysis we aggregates the processing speeds of Rabin's Mystic secret sharing scheme with a single cloud provider GoogleDrive on different share sizes of 1kb, 10kb, 100kb, 1mb and 10mb and also we found the aggregates of processing speed in Multi-cloud providers, for this scenario, we implemented on openstack swift object storage with a ring environment on different share sizes and the fig. 5 depicts the scenario. Next we analyzed openstack performance measure with increasing file size, network delay on increasing partition size, access structure of key pair with the nodes along with these 3 factors, we grouped the values of bandwidth used, network tolerance by openstack, key pairing with the nodes and online zone access in swift object storage. We found these 3 factors grouped and splined with values with the instances created on openstack like 2^o,3^o, 4^o,..$9^o$of virtual object storage(VO) from fig.6. we concludes that access structure with the openstack performance, our scheme are reliably faster.

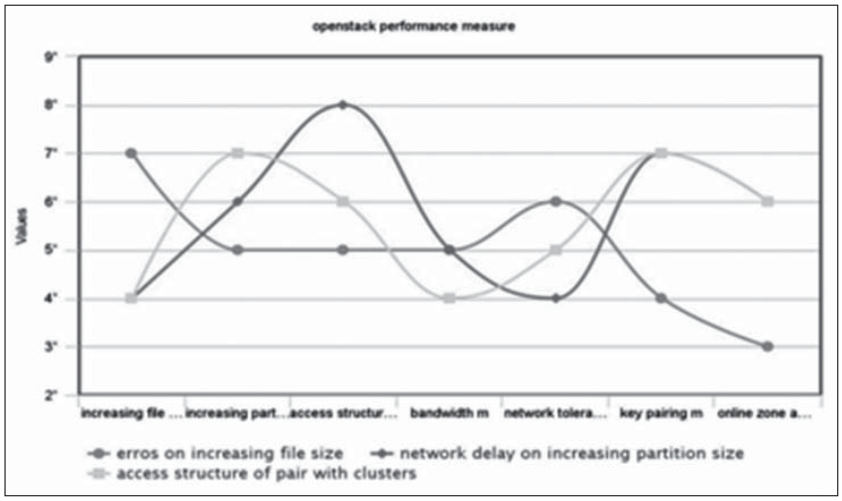

Figure 6. Openstack performance measure

After we analyzed with kaavo-IMOD Jira, a Multi-cloud management tool with an AES 256 bit data encryption and we splits the file shares and repositories over multiple virtual server that serves as multiple cloud service provider. In kaavo we can access CSPs like IBM,AWS, Rackspace, openstack and all of the market big players private, public and hybrid cloud service provider. This kaavo provides the benefits of Infrastructure as a Service (IaaS) with increased flexibility-pay_as_you_go model, lowers cost, reduction in time to market-available within minutes of signup and generate access to critical IT resources. With our Rabin's Mystic sharing scheme, we implemented and cloned our system with kaavo-IMOD, all the virtual servers have properly configured the firewalls. Secure VPN connection for sending and receiving file(share) between the internal server/datacenter/virtual servers in public clouds, from the kaavo-IMOD we monitor the rule based alerts and monitoring of CPU, disk, bandwidth, memory, kaavo clones our system and bring it to application service centric n-tier configuration i.e., automation to bring entire system(single and multi-clouds) services for one application within a click and brings the ability to access your information from anywhere in the world.

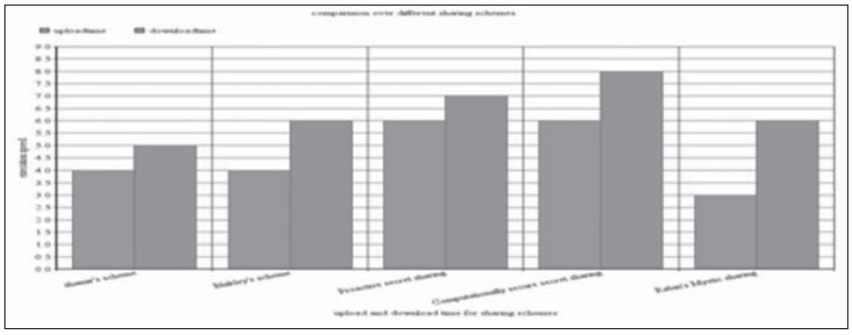

Figure 7. Comparison over different sharing schemes

Here we analyzed different secret sharing schemes[7] with our system, the execution speed of our scheme was better than the shamir's, blakley's, proactive, computationally secret sharing schemes in terms of execution speed while uploading and downloading the secret shares from cloud servers. Next we analyzed the throughput measure over Real time bandwidth monitor with the parameter time and size of the share, our system connected to single cloud and multi-cloud servers in a bandwidth of 100mbps and running time window for 1 hour calculated based on cloud servers when they are accessed in traffic, using this factor we calculated for a16mb file our system splits the share and it uploads to single server takes 0.59s and 0.65 for multiple CSPs, for a 32mb it takes 0.67s on single cloud and 0.74s on multiple CSPs and we shown this on fig.8 that shows the clear representation having differences with single cloud and multiple CSPs and this may vary on different configurations.

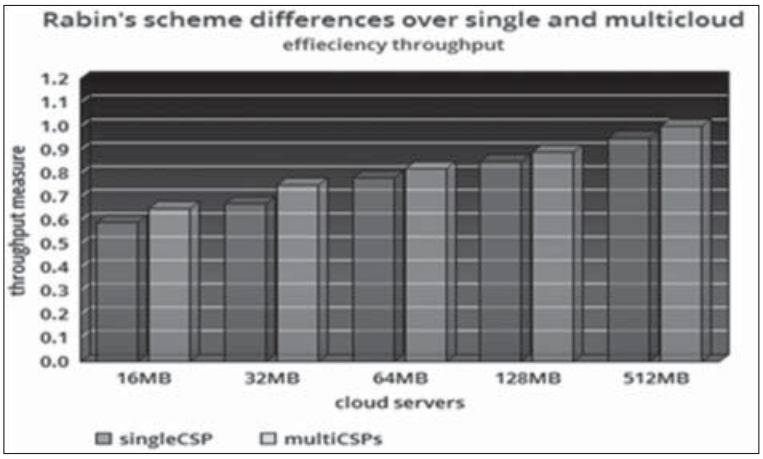

Figure 8. Rabin's scheme differences over single cloud and Multi-cloud

We analyzed cloud servers performance over kaavo-IMOD with different parameter for showing how cloud servers are utilized by the user with different CSPs, and also we calculated the percentage of parameters like average time to provision a node, ie., how much processing Virtual Objects (VO) are used, VRAM size and processor speed of each VO instance and also we consider time taken to deploy an application and in case of data/share upload and download and we computed performance over the percentage of encrypting the file share, traffic analysis and percentage of managed node on storage, while increasing the user, our system suffers with the processing capability, as we need much more VO instances for handling many number of users. We monitored using iperf tool and analyzed with cloud server virtual images like Amazon EC2, Google Drive, IBM, Rackspace cloud, Expedient and openstack cloud, from our conclusion we found Amazon exhibit 16-48% higher memory performance, Rackspace exhibited 16-27% higher Vcpu performance and Expedient achieves over 66% of higher storage performance. From all of these discussion we finally shows performance indication for cloud servers with all the factors included and fig.9 depicts the summary of performance of individual cloud servers for our system.

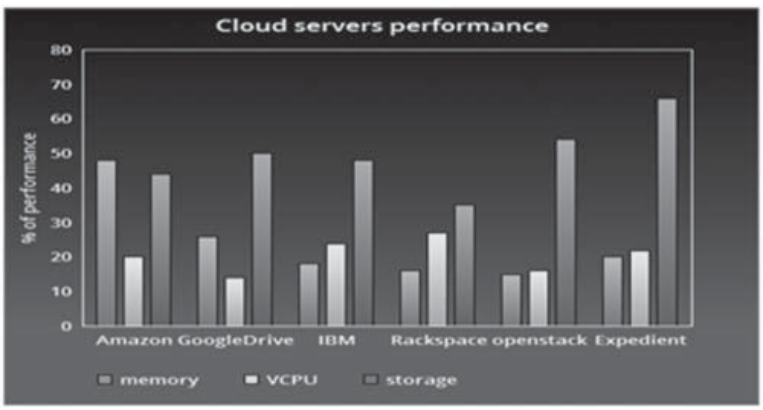

Figure 9. Performance Indicator Over Cloud Servers

6. CONCLUSION

In terms of security concerns in cloud computing, our RABMYST scheme was designed to provide security and hassle free access to the CSPs. We evaluated our scheme with different performance measures like increase in file sizes, key pairing structures, bandwidth factors and network tolerance in openstack a model for accessing the secured Multi-clouds. We provide the security and access to Multi-clouds interface kaavo-IMOD with the analysis on different secret sharing schemes while uploading and downloading a file and we analyzed throughput measure with different file sizes over single cloud and Multi-cloud provides. We also estimated with VO instances, Vcpu and storage performance on different cloud servers, our scheme shares the file not key, here we avoid key management issues, computation overhead and improved performance to access the cloud storage servers without any complexity.

References

1. Fahad Alsolami et al., CloudStash: Using Secret-Sharing Scheme to secure data, Not keys, in Multi-clouds, IEEE, 11th International conference on Information Technology: New Generations Apr.2014, pg 315-320.
2. Ambikavishalpawar, et. Al., Design of privacy model for storing files on Cloud Storage , Journal of Theoretical and Applied Information Technology, 20th September 2014, vol 67, No.2.
3. Giuseppe stecca, Antonio puliafito, A Cloud based system to protect against industrial Multi-risk events, 48th CIRP conference on manufacturing systems 2015, Elsevier ProcediaCIRP41(2016) pg.650-654.
4. Seung Hyun seo, et. al., An Efficient certificateless Encryption for secure Data Sharing in public clouds, IEEE Transactions on Knowledge and Data Engineering Vol.26, No. 9 sep.2014.
5. Bruno opera et. al., Cloud Collaboration for Forensically Ready Cyberspace, International Journal of Scientific and Engineering Research Vol. 6. Issue 5 May 2015.
6. M. Muhil, et. al., Securing Multi-Cloud using secret sharing Algorithm, 2nd International Symposium on Big Data and Cloud computing ISBCC'15, procedia computer science 50(2015) 421-426, Elsevier, Science Direct.
7. S. Jaya Nirmala, et.al., A comparative study of the secret sharing Algorithms for secure Data in Cloud, International Journal of Cloud computing: Services and Architecture. Vol2, No.4, Aug.2012.
8. PhillippeBeguin, et.al., General Information Dispersal Algorithms, Elsevier, Theoretical Computer Science 209(1998) pg. 87-105.
9. NesrineKaaniche, et.al.,CloudaSec: A Novel public-key based framework to handle Data Sharing, 11th International Conference on Security and Cryptography, SECRYPT 2014, at Vienna, Austria. Aug.2014.
10. Mingqiang Li, On the Confidentiality of Information Dispersal Algorithms and their Erasure Codes M Li - arXiv preprint arXiv:1206.4123, 2012.
11. Rafe al M.de O. Libardi, et.al., MSSF: A step towards user friendly Multi-cloud Data Dispersal, 2014 IEEE 7th International Conference on Cloud Computing.
12. Andrew Tytula, 237022, Peer-to-Peer File Sharing system using an Information Dispersal Algorithm, 95.495, Honours project, Carleton University.
13. Sian-Jheng Lin et.al., An Efficient (n,k) Information Dispersal Algorithm based on Fermat Number Transforms, IEEE Transactions on Information Forensics and Security IEEE Transactions on Information Forensics and Security (Volume:8, Issue: 8) pg. 1371 - 1383
14. Tatiana Ermakova, et.al., Secret Sharing for Health Data in Multi provider clouds 2013 IEEE 15th Conference on Business Informatics 15-18 July 2013 pg. 93 – 100.
15. Manvi Mishra, et.al., An Assessment of cloud computing: Evolution, IJRET, eISSN:2319-1163.
16. Kota Tsuyuzaki and Masahiro Shiraishi, Recent Activities Involving openstack swift, " Regular Articles", NTT Technical Review, Vol.13, No.12, Dec.2015.
17. Alexander Adamov, Security Best Practices version 1.0, Mirantis Openstack, 2015.
18. Lorin Hochstein, Lead Architect, Cloud services Nimbis Services, Python APIs: The Best-Kept Secret of Openstack, IBM "Developer Works" 19th June 2013.

19. Michael O. Rabin, Efficient Dispersal of Information for security, Load Balancing and Fault Tolerance JACM, April, 1989.
20. Hugo Krawczyk, Secret Sharing made Short, Springer-Verlag, 1998.
21. Rui Zhao, et.al., SafeSky: A secure Cloud Storage Middleware for End-user Applications. Reliable Distributed Systems (SRDS),2015, IEEE 34th symposium on .pg. 21-30.
22. Zhen Chen et.al., Cloud Computing- Based Forensic AnalysisforCollaborative Network Security Management System, Tsinghua Science and Technology. ISSN:1007-0214, 05/12 pg.40-50, Vol. 18, No.1, Feb2013.
23. Josiah Dykstra, et.al., Acquiring Forensic evidence from infrastructure-as-a Service Cloud Computing: Exploring and Evaluating tools, trust and techniques, Elsevier, ScienceDirect, Digital Investigation 9(2012) s90-98.
24. Josiah Dykstra, et.al., Design and Implementation of FROST: Digital Forensic Tools for the Openstack cloud computing platform The International Journal of Digital Forensics & Incident Volume 10, August, 2013 Pages S87-S95.
25. Eoghan Casey, Digital Evidenceand Computer Crime, 3rd Edition, www.elsevierdirect.com/companions/9780/23742681.
26. Yuan Zhang et.al., SCLPV: Secure Certificateless Public Verification for Cloud-Based Cyber-Physical Social Systems against Malicious Auditors, IEEE Transactions on Computational Social systems Vol.2, No.4, Dec.2015.
27. Roberto vigo, et.al., Automated Generation of Attack Trees, IEEE 27th Computer Society DOI 10.1109/CSF 2014.31 pg.337-350.
28. www. Openstack.org.
29. Gaidhankar Sushil, et.al., Multi-cloud storage using Shamir's Secret Sharing Algorithm IJAETMA Vol.1,Issue 7 Dec.2014. ISSN:2349-3224.
30. https://metacpan.org/prod/crypt::IDA.