International Journal of Combined Research & Development (IJCRD)
eISSN:2321-225X;pISSN:2321-2241 Volume: 6; Issue: 5; May -2017

S^3-IDC: Technique of Secure Splitting & Storage of Information in Multiple Data Centers

Vijay G.R
PhD Scholar, Dept of Computer Science and Engg.
JNTUA
Anantapur, Andhra Pradesh
E-Mail: vijay.gcr@gmail.com

Dr. A. Rama Mohan Reddy
Prof.: Dept. of Computer Science and Engg
SVU College of Engineering
Tirupathi, Andhra Pradesh
E-Mail: ramamohansvu@yahoo.com

***Abstract*— With the increasing pace of adoption of cloud for storage services, there is also an increasing vulnerability of the data being stored in the cloud. The cloud servers are not only used for storing the data but it is also used for collaborative sharing of the data by numerous customers, which leads to higher prone of data insecurity over the cloud servers. In the past decade, majority of the research attempts were found to adopt conventional cryptography technique with less focus on its sustainability as robustness. Hence, we present a novel technique called as S^3-IDC (Secure Splitting and Storage of Information in Data Center), which performs a unique splitting of data, carries out encryption using AES 128, and stores the encrypted chunks of data randomly over various rack servers on numerous cloud zones. The outcome of S^3-IDC is found to outperform existing security standards to show the effectiveness of proposed system.**

Keywords-component;, Cloud Storage, Data Security, AES 128, Data center, Cloud Server

I. INTRODUCTION

With the increasing use of the various enterprise application and mobile computing, cloud computing has been evolved up as a boon to the user who desperately seeks solution to the storage system. Cloud computing a form of highly distributed computing system that allows higher extent of pervasiveness to its user [1][2]. Various significant services offered by Cloud Computing are SaaS (Software-as-a-Service), PaaS (Platform-as-a-Service), and IaaS (Infrastructure-as-a-Service) [3]. The Cloud computing technologies are grouped into 4 sections which includes

- **Software as a Service (SaaS):** is an on demand application service for cloud computing.
- **Platform as a Service (PaaS):** is an on-demand platform service to the host customer Application.
- **Data Storage as a Service (DSaaS):** is an on-demand storage service.

Infrastructure as a Service (IaaS): is an on demand infrastructure delivery services. From storage viewpoint, the cloud computing offers highly reliable storage services from its data centers, which comprises of various forms of highly configuration cluster with extensive storage viability. There are various advantages of adopting Cloud as storage services as it offers storage with additional services of data security in much cost effective prices as compared to conventional storage system based on physical servers [4]. Cloud Security Authentication System involves a large amount of users who can get connected with a distributed or multi-cloud environment where they can upload files and store them without apprehension of large amount of space requirement. Cloud Computing technology is designed as a service oriented, internet based, secure and suitable data storage computing technology which provide the ease of on-demand network access to shared pool of computing resources , the cloud computing technology connects software , hardware , infrastructure and data storage capacity to provide a configurable system with various services over the internet. Cloud service providers deliver the distributed shared applications which can be accessible from any browsers, Desktops or mobile Apps.

One problem that has disallowed their emergence is the security issues associated with the data. In order to maintain a secure services mechanism cloud authentication systems can develop some techniques which will be useful for the secure management of stored data. Cloud Authentication system extemporize different password techniques but the major issues related to the textual password authentication is it is very much vulnerable to the brute force attack and it is very easy to break the password and an intruder can easily retrieve the data. It is been observed that the multi-dimensional password authentication techniques also have some drawbacks but the multidimensional password authentication system for securing data reduces the probability of brute force attacks. According to the proposed Encryption and Decryption techniques a user can upload and maintain his data on the cloud application with the following features e.g. *Ease of use*, *Security*, *Anonymity*, *Performance*, *Robustness*. There is a growing need for an effective, efficient Cloud Security Authentication Technique for Secure, reliable, data storage services. Current secure data storing approaches in cloud computing technology do not scale to such a domain. Most existing Authentication technologies proposed or prototyped to date suffer from problems with security, lack of obscurity and performance. This paper introduces a technique to solve the security issues over data storage in cloud. Section 2 discusses about the prior studies

followed by problem identification in Section 3. Section 4 discusses about the proposed system followed by illustrative discussion of research methodology in Section 5. Section 6 discusses about the implementation followed by result discussion in Section 7. Finally Section 8 summarizes the paper.

II. RELATED WORK

Study on security of the data storage has witnessed numerous research publications most recently. In order to strengthen cloud storage security, it is important that SLA (Service Level Agreement) should be adhered. Notable work done by Popa et al. [5] has emphasized on the SLA by introducing a framework called as *CloudProof*. The framework allows the customer to identify the various forms of violations towards data integrity on Amazon S3 cloud services. The authors have studied the framework using performance parameter of storage overhead and processing time. Similar form of framework to identify the violations was also proposed by Tang et al. [6]. The authors have introduced a framework called as FADE using conventional cryptography techniques. The framework is found capable of detection of file deletion on cloud storage and is tested on Amazon S3 cloud. The authors have studied the effectiveness of FADE using rate of data transmission and processing time required for key management.

Ren et al. [7] have proposed a technique of secure data accessibility on mobile application in cloud computing. The study has introduced an encryption policy to maintain the integrity of the data during uploading and downloading the file from the cloud servers. However, the study provides less evidence of its outcome. Study towards security over data sharing was presented by Dong et al. [8], where the authors discussed their security algorithm using re-encryption scheme. The outcome of the study is found to support identity-based encryption scheme as well as public key encryption scheme at same time. The technique performs transformation of the encrypted data of the customer into encrypted text for rending data security. Bessani et al. [9] have introduced a framework called as DEPSKY for enhancing the data confidentiality, availability, and integrity. The authors have performed the experiment using PlanetLab application on commercial cloud.

Another technique to perform secure data storage is to create a data deduplication policy. Study carried out by Stanek et al. [10] have proposed such scheme where the authors have designed an encryption technique for ensuring semantic security for the data. The simulation study was performed on C++ application considering AES algorithm of 256 bits and SHA algorithm of 256 bits. Kim et al. [11] have presented a framework called as Unity which is completely focused on solving the data availability issue over cloud. The experiments were carried out on Xen hypervisor and compared their performance with that of Dropbox. Cao et al. [12] have presented a technique for verification of public integrity using third party emphasizing on the reliability factor of data using LT (Luby Transform) codes. Murthy [13] have presented a secure authentication process using cryptography using homographic encryption approach with multiple key distribution centers. The authors have worked on key generation system using Paillier cryptosystem that authenticates the user accessing cloud storage. The technique was also compared with existing 3DES algorithm as well as symmetric key-based techniques. However, the study is not found with elaborated result discussion for which reason it is quite difficult to ascertain the claims of the authors.

Xiong et al. [14] have presented a framework called as *CloudSeal* for ensuring the secure sharing as well as distribution of the confidential data using cloud storage. However, CloudSeal is only focused on confidentiality of the data and is implemented over Amazon Web Services as well as CloudFront. CloudSeal perform two steps of encryption where the initial level of encryption is done using symmetric scheme and secondary level of encryption is done using proxy-based ciphering scheme.

Gasti et al. [15] have focused on privacy issues of data storage and introduced a concept of *deniable cloud storage*. The implementation of the concept has been done for sharing system too where security is rendered using available tools like TrueCrypt. Kamara et al. [16] have presented a framework called as CS2 emphasizing on data integrity, confidentiality, and verifiability. Kaaniche et al. [17] have used identity based cryptography for cloud storage for furnishing data secrecy as well as seamless data access. Vimercati et al. [18] have presented a technique to secure resource and authorization management system for cloud storage. Study on data leakage problems have been carried out by the Xu et al. [19]. The author used the approach of data deduplication technique to provide confidentiality of data. The authors have designed the programs in C using SHA256 and AES algorithm on Linux machine. The outcome of the system is mainly evaluated using processing time. Similar strategy of deduplication scheme was also seen in the work of Puzio et al. [20]. The authors have presented a framework by name *ClouDedup* that provides secure storage facilities on block based of data. The framework also provides a typical key management scheme. The outcome of the study was evaluated using storage space overhead, where the security is achieved using SHA256 hash.

Standard work towards sharing the sensitive data over cloud was seen in the literature of Thilakanathan et al. [21]. The authors have discussed some of the significant factors that should be considered at the time of sharing the data. Margret [22] have presented a technique that enables security of data sharing process for multiple owners using attribute based ciphering process over private cloud. Damgard et al. [23] have presented lightweight protocols for higher ranges of data security using key management over cloud environment. Tysowski and Hasan [24] have presented a re-encryption technique to be governed by the service provider of cloud.

However, the system depends closely on the managers who will be responsible for generation of the keys.

Hence, it can be seen that there are maximum number of the literatures in last 5 years that has focused on data security over the storage systems on the cloud. However, all the attempts were done considering conventional cryptography protocols or deduplication methods where there is a strong dependency on key management. The pitfall in the existing system is very few works have emphasized on performing non-conventional encryption policy. Moreover, once the data is uploaded it is found to store on single datacenters over multiple clusters, showing ineffectiveness of data indexing mechanism that leads to critical vulnerability of data ownership.

III. PROBLEM IDENTIFICATION

Security has always been the great concern in cloud computing. The existing service providers for cloud storage are not found to provide the optimal security and data privacy which is originally promised to the customers. Some of the facts that have been identified in the security issues in data storage over cloud are discussed as follows:

- **Data Leakage**: This issue arises where the secondary copy of the original data exists over the server posing higher ranges of vulnerability to the ownership of data. One of the issues of data leakage was found most recently by Dropbox [25]. It was found that various confidential files that have been deleted by the user in Dropbox still exist. Another significant discovery is that various confidential metadata about the history of the user is also possible to find. Hence, this problem could be solved using encryption; however, encryption is not the only solution in public cloud. This problem could be solved if the data storage, indexing, and key management are done properly.
- **Vulnerability of Public Storage**: In the existing system e.g. Google Drive, Microsoft, Dropbox, it is feasible to store any forms of data. Such data are stored in unknown proprietary storage known only to the service providers. In majority of the cases, the storage facility is shared which renders more vulnerability of customer's data. It is feasible for the illegitimate member in the shared storage to access the credentials of an account. Therefore, usage of strong encryption protocols like AES doesn't really help out in securing the data. Moreover, as the cumulative data is stored in one particular data center (single cloud), the situation turns worst. This problem could be solved by storing the data on multiple data centers and adoption of a non-conventional file indexing system could really help.
- **Defective Encryption Policy:** Another significant issue observed is the usage of the defective encryption policy over cloud storage. It should be known that cloud storage is not merely for storage solely as various customers like to use it for secure sharing and data collaboration too. Surprisingly, it has been noticed that when there is an attempt to share the confidential data among the multiple users on single cloud, it is more prone for intrusion by the technical members of the service providers of storage in data centers only. Hence, there is no guarantee of ownership, privacy, confidentiality, integrity of the data. This problem could be avoided if a system is developed that doesn't give any accessibility of secret key to any illegitimate user.

Hence, all the above mentioned points are the problems that have been identified in the proposed study. Therefore, the problem statement can be given as – "*It is a computationally challenging task to design a cost effective framework of secure storage on multiple cloud without much proneness of accessibility of keys by any illegitimate member*." The proposed system mainly attempts to investigate the solution towards this problem statement and has evolved up with a technique that can ensure adherence to the maximum security standards of the data storage much complexities being involved in it.

IV. PROPOSED SYSTEM

The prime purpose of the proposed system is to design a secure storage and accessible framework in cloud computing that can offer greater deal of privacy, confidentially, and integrity. We have coined the name of the framework as S^3-IDC that stands for Secure Splitting and Storage of Information in Data Center.

The proposed system highlights a secure cloud storage management for safeguarding the data which are uploaded by the users in the cloud application. Cloud security architecture defines a multi cloud security structure where an authenticated user only can upload the data in a distributed cloud system and keep it safe for his future requirement. As now a day's huge amount of data requires lots of space and that requested amount of space cannot be provided by any of a single Personal Computer which has a very less amount of space. And the computing resources also have very limited amount of processing power which would be responsible for disk overhead. The uniqueness of this proposed cloud security architecture is it uses a very distinctive encryption and decryption mechanism with key distribution technique to secure a large amount of data with anonymity. Proposed system share a large amount of services to protect the information from any types of vulnerable attacks. As there are various proposed authentication and security associated with data maintenance techniques have been implemented till date but it has been observed, there are very less attempted techniques to secure the cloud storage management.

The main aim of the research work to design a cloud security architecture which can be developed to maintain the data security of various cloud applications and to authenticate the users from performing any types of illegal activity the objectives of the proposed system are e.g. i) *Verification*: The system should allow online users to get authenticated and verified by the cloud application interface, ii) *Secure Data Upload*: The system should generate some splitted keys to secure the data which have been uploaded by the genuine users., iii) *Distributed Key Mechanism*: the system will provide

a distributed key mechanism to secure the uploaded data. The system will provide distributed ciphers for encryption and decryption of the uploaded and downloaded data respectively. The prime contributions of the proposed framework are as follows:

- To design a common *application interface* that interfaces global users with existing data centers.
- To develop a *Synchronous Server* that can assist in data management system and simultaneously acts as a bridge between the user and the cloud systems.
- To allow the system to perform *splitting* of the user's data and store it in multiple and anonymous rack server over distributed cloud environment.
- To develop a *lightweight cryptographic algorithm* using AES that can perform encryption of the splitted data of user and reposit it randomly over the distributed clouds.
- To evaluate its effectiveness with existing security protocols.

V. RESEARCH METHODOLOGY

The architecture of the proposed system is shown in Fig.4. The design principle of the architecture are mainly governed by two significant actors viz. i) customers and ii) autonomous administrator. The customers are the initial actor of the proposed system that seeks to use the system for storing the information. The information to be stored on cloud could be of any forms. The autonomous administrators are responsible for the data management. In order to explicitly understand the working principle of the proposed system, let us look closely to the components used in the design process:

- *Application Interface*: A common application interface is created using Java that allows all the essential actors (mainly customers) to upload as well as access their information. Fig.2 shows the schema of application interface that connects the online customers with the cloud storage.

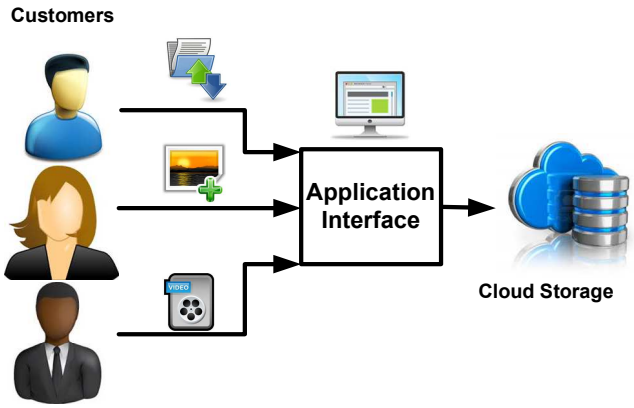

Figure 2 Schematics of Application Interface

The application interface initially allows the user to undergo a secure authentication policy by validating the user ID and the confidential passwords of the customers. Once the validation of the user profiles are done than customers will be allowed to upload as well as access their personal information that they choose to reposit on the cloud storage. Every time the user will require either uploading or accessing their stored data, it has a dependency on the next component called as Synchronous Servers.

- *Synchronous Servers*: One of the interesting parts of the proposed design principle is that the original data uploaded by the users will not be directly reposited on the cloud storage. The proposed system introduces a middleware system called as Synchronous Servers that creates its own file systems and stores sequences of the customer's data (refer Fig.3). The synchronous server is responsible for the autonomous data management system for multiple customers on a given cloud. The Synchronous Server is required to maintain proper indexing of the customer's file. Once the file is processed through Synchronous Server, the system will further perform two simultaneous operations e.g. i) performing encryption on the customer's data and ii) communication with the data centers. As direct communication through the data centers are sometimes not feasible in many geographical regions, hence, the system choose to take the assistance of availability zones.

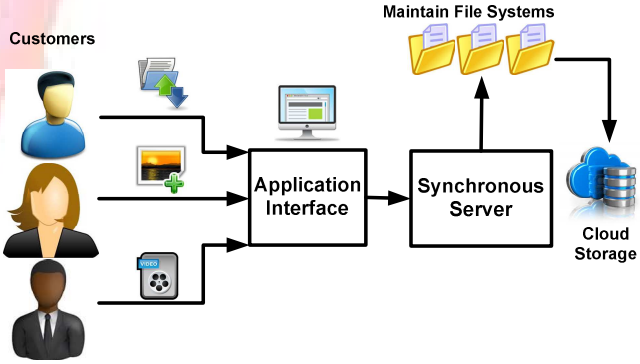

Figure 3 Schematics of Synchronous Servers

The outcome of the Synchronous Server component will be the indexed file with a specific file formats with an array to store the ordering of the information uploaded or requested by the online customers. After the data is processed by this component, it is subjected to encryption using lightweight cryptography.

- *Lightweight Cryptography*: The term *lightweight* will refer to adoption of such a cryptographic technique, which is not only faster in performing both encryption/decryption, but also occupies less memory. However, we will not directly subject the processed data of Synchronous Server for encryption. The processed data from Synchronous Server is converted to binary values, which will lead to generation of a 128-bit secret key. The next step is to perform

encryption of the binary data with AES algorithm. We choose to implement AES as i) it supports faster computation and ii) it is less vulnerable to majority of the lethal attacks on internet with 64 bits. After carrying out the encryption using AES algorithm, the system leads to the encrypted version of the binary data, which will be further subjected to data splitting process.

- *Data Splitting*: This component is responsible for splitting the data to the different data centers based on the availability of the rack servers. The core system connecting to the application interface also maintains a matrix for the data centers with the availability of the rack servers on the cloud. The system then splits the processed data of all the gross active customers based on the number of the availability of the rack servers. This process of splitting the data will mean that if there are 10 petabytes of the processed data, the data will be splitted as 3petabytes, 2 petabytes, 1 petabyte, 3 petabytes, and 1 petabyte, if there is availability of 5 rack servers for time being.

One of the uniqueness in the proposed system is that the generated secret key that is encrypted with AES algorithm is stored randomly on the designated rack servers that are found to be available on that time instance. The system also providers better service by storing the secret key in the network, which is completely unknown to the user as well as administrator. Hence, the proposed system performs storage of the customer data in highly distributed and secured manner.

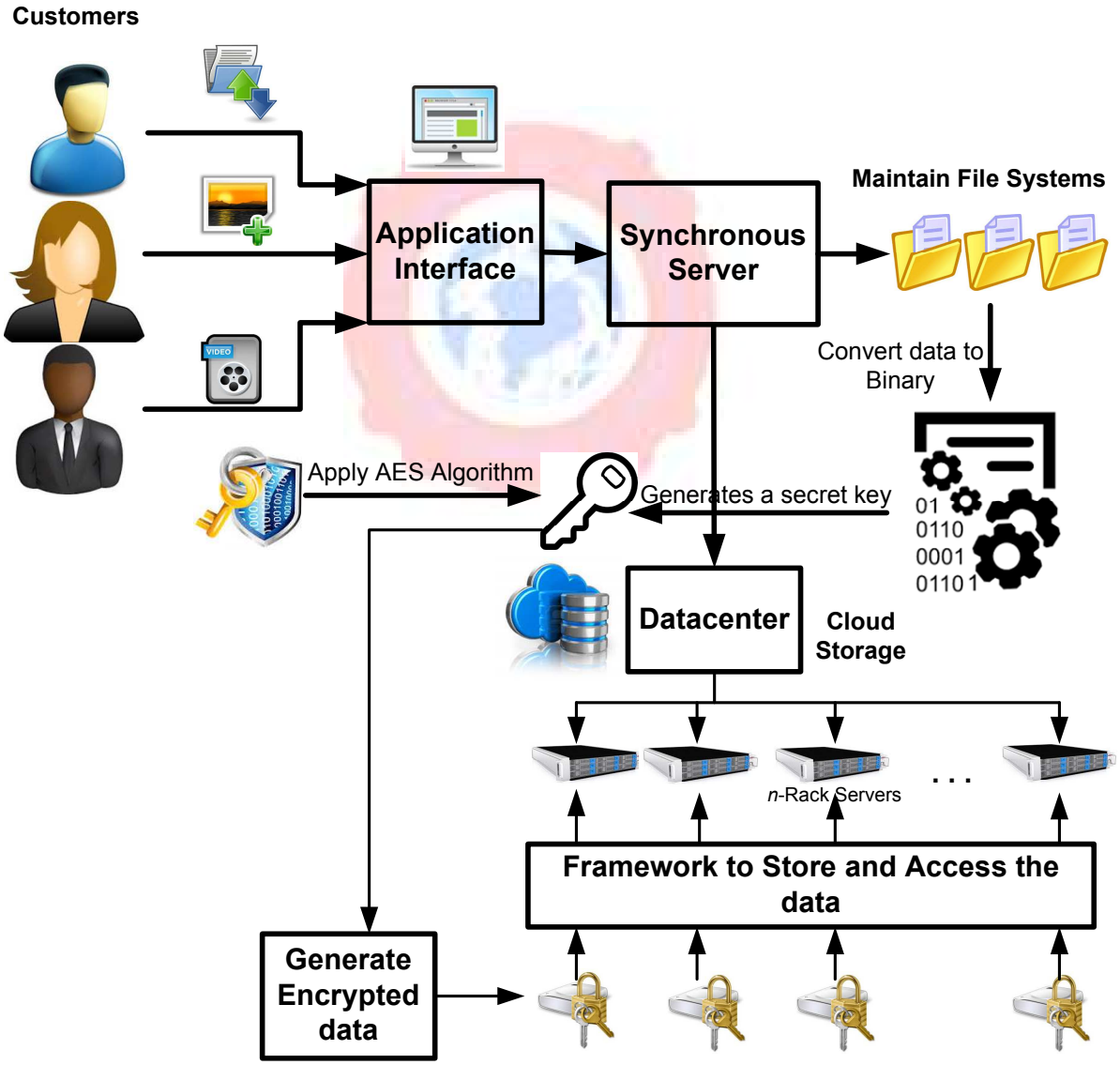

Figure 4 System Architecture of Proposed Project

VI. IMPLEMENTATION

The implementation of the proposed system is carried out on 34 machines with multiple configurations. All the machines are of different operating system, processor speed, and memory with pre-installed JDK environment for running the application. We choose to develop it in Java environment as it will be feasible to check the real-time evaluation of the proposed system and more over multiple machines of heterogeneous configurations can be checked at same time. The prototype development of the S^3-IDC has been carried out in highly distributed manner in MyEclipse. Out of 34 machines, 5 machines were considered for autonomous administrator access, ten machines for running the cloud-server, and remaining 19 machines were for distributed cloud customers. All the customers underwent a profiling stage that considers their personal and contact details for enrollment process. A database has been set up in SQL which maintains the credentials of the customers and once the customer attempts to access their account, the system uses MD5 algorithm to secure the password. However, we assume that the initial credentials are never safe and it is meant for only initial access to their priviledge application. The rack servers were hypothetically designed over the considered machines on its available storage. A specific threading mechanism in Java as well as algorithm calls ensures the suitable operation of the file indexing mechanism of the proposed system. The proposed system uses an encryption technique, which becomes mandatory for the customer to undergo it whenever they are attempting to upload their data over the cloud storage. An algorithm is designed for this purpose that considers the inputs as the request of the customer to store their data over the multiple cloud storage system and then after processing ensuring storing the encrypted chunks of data randomly on the available cloud without being any preemptive knowledge about the location of the file or the secret key.

Algorithm for Secure Splitting and Storage of Information in Data Center

Input: Request of Customer for Storage

Output: Successful storage of Encrypted data.

START

1. Initiate *db* connection from user to application interface.
2. Use MD5 to secure the password
3. Validate the user ID and password of customer.
4. Convert the data to binary
5. Generate a Secret key
6. Encrypt the data with Secret key.
7. Apply Rijndael key Schedule & derive RoundKey
8. BitXOR(state of byte, Block(RoundKey))
9. Apply Non-linear substitution step
10. SubByte: oldByte→newByte(LookUpTable)
11. ShiftRow: Transpose(state of Row)
12. MixColumn: Mix(state of column)
13. Add all the RoundKey
14. Check the size of data
15. Evaluate remainder=size % cloud no
16. Estimate the size of split data (S)
17. (Size of data-remainder/cloud no.
18. Last data size=S + remainder
19. Insert key values to random cloud

END

The above algorithm takes the input of request of a customer to store a specific set of data and checks the connectivity of the application interface with the storage servers over cloud. After validating the user ID and initial password, the system converts the customer data to binary. This task is also accompanied by generation of secret key. The system applies AES algorithm of 128 bit to further encrypt the secret key. Therefore, the algorithm posses the capability to furnish AES 128-bit encryption for both static as well as dynamic data that gives a wide supportability for the users to maintain their private data, secret keys, and most important, it offers ultimate data security even if any chunk of the encrypted data is being compromised. According to the algorithm, it stores the chunks of encrypted data randomly on the available rack servers, without any physical storage of key and location information. Hence, S^3-IDC offers higher scalability as it is completely independent of any file types. The algorithm can cipher any forms of data of customers with less complexity of the management of keys. The technical adoption of the algorithm is quite higher as it can be simply integrated on the exiting data storage system without much re-engineering activities. Moreover the space complexity is less owing to iterative use of same key size of 128 bit.

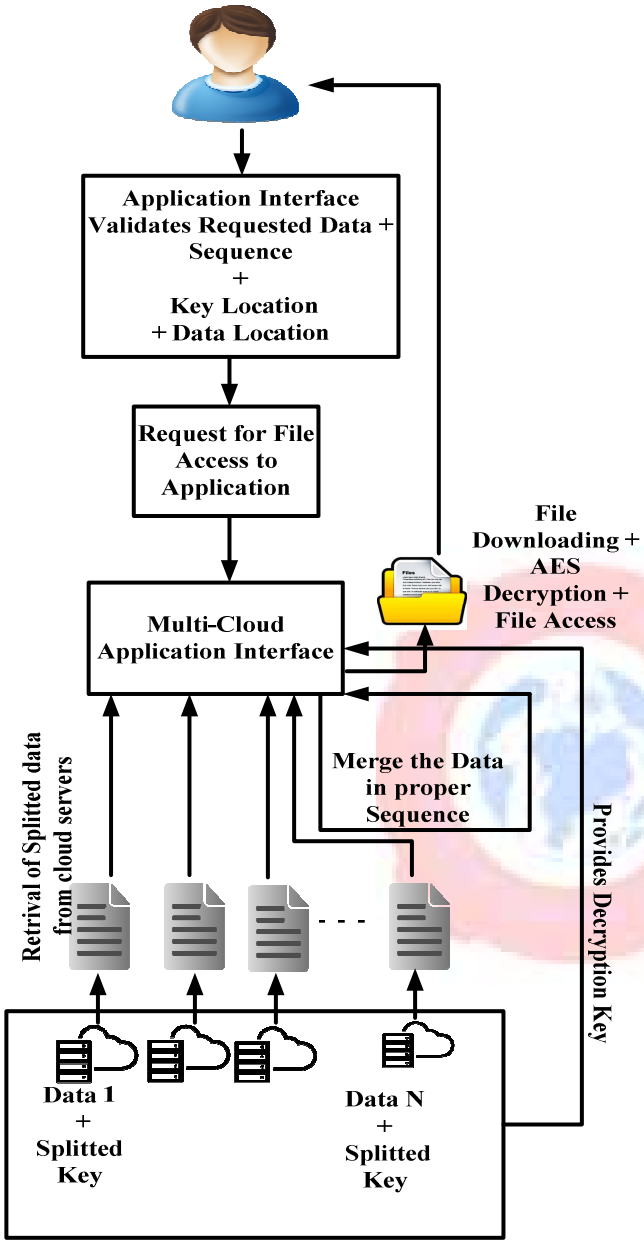

Figure 5 Steps for Decryption and Accessing Encrypted File

The decryption stage of the algorithm is pictorially shown in Fig.5, which is just reverse of the encryption algorithm. Hence, it can be said that S^3-IDC resists even the autonomous administrator, webmasters, root or any illegitimate malicious access to the confidential encrypted data. Even if the encrypted data is compromised, it is impossible to obtain the complete data as the rest of the data is maintain on different rack servers whose location is unknown. Even if the encrypted data is clustered maliciously by intruder, they will never have an access to the key at same time. The algorithm is nearly impossible to break or perform cryptanalysis. Hence, S^3-IDC offers a cost-efficient, lightweight cryptographic technique that can perform encryption for the data to be stored on multiple cloud and also furnishes a faster accessibility of data without any significant impact on the networking performance of cloud.

The implementation of the discussed algorithm shows that there is a typical pattern of processing the data once the information is indexed by the Synchronous Server. Fig.5 shows that the proposed algorithm for encrypted data storage is governed by customers as well as autonomous administrator. Finally, the storage of the splitted data is carried out and stored randomly over multiple rack servers in highly distributed fashion. The advantage of this process is lack of knowledge of storage location by any of the users as well as any administrators in the cloud environment, which confirms highest forms of data privacy.

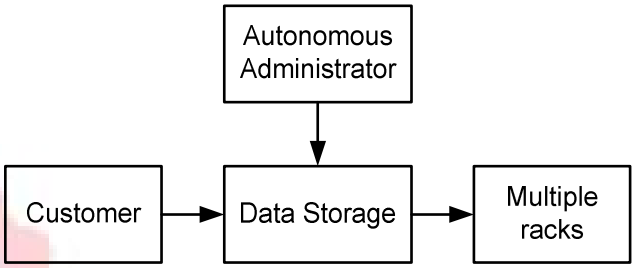

Figure 5 Schema of Gross Process

Fig.5 also showcase that data storage of the proposed system is configured and managed by autonomous administrator, while customer shares their data for storage purpose, which is further stored in different locations of data centers on multiple clouds inspite of single cloud in existing system.

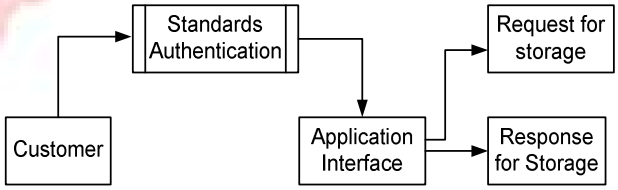

Figure 6 Accessing Application Interface

Fig.6 shows that the customer will be required to perform standard authentication process to validate them online using the common application interface. The authentication process is carried out by customers by furnishing their initial credentials in the forms of user ID along with the secret password to be fed in the application interface. It should be noted that we are also using an application interface common for accessing rights for autonomous administrator too. Once the authentication is successfully achieved, the customer will be permitted to share their data of any types in the cloud by using request for storage and response for storage process. The system allows all the cumulative data to pass through the Synchronous Server that maintains the sequences of data of the customers. The system ensures that all the data that are shared by the users reaches cloud storage location on various racks. However, we choose to initially find the availability of the rack servers on multiple clouds and choose to randomize the location of the racks in order to maintain non-anonymity of the data as an integral part

of security system. This part of the process is beyond the control of either customer as well as autonomous administrators of the cloud service providers. Hence, the system can be said to provide an ultimate data privacy, confidentiality, and integrity too.

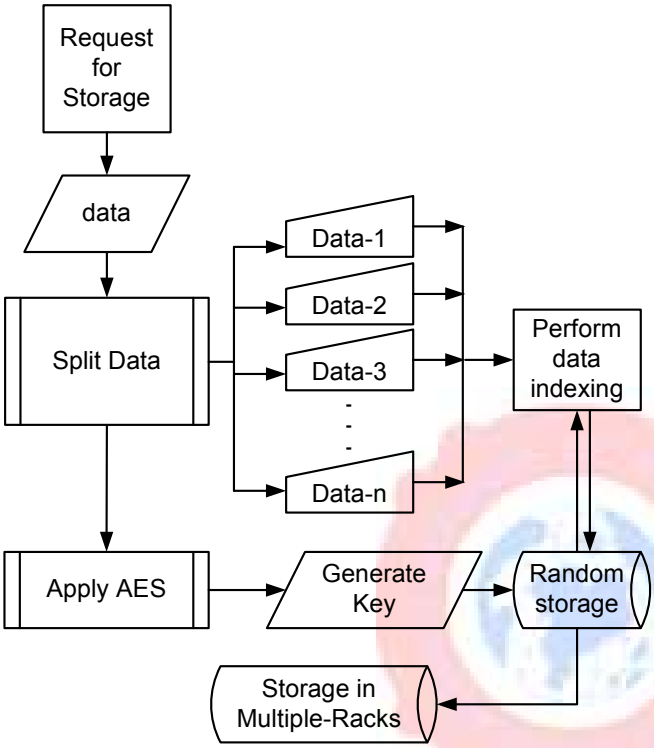

Figure 7 Schema of Encrypted Data Storage

Fig.7 shows that once the data arrives on the system in the form of request message from the client to store the data, the system performs splitting the data depending upon the number of available rack servers on multiple data centers. One interesting point is the system searches for the availability of the rack server in highly encrypted pattern, which upon completion gives the number of available rack servers and not the location information of the rack servers. This process ensures that once the data is stored in any of the rack servers, only the system knows how to find it. Hence, the splitting of the data is carried out based on number of freely available rack servers. In case the number of splitted data doesn't match evenly with number of available rack servers, the extra quantity of the data is always stored randomly on any of the available rack servers. All the data that are splitted is also indexed followed by the conversion of the data to the binary values for generation a secret key. This secret key is again subjected to 128 bit AES algorithm to further generate encrypted data. The secret key is stored randomly on any of the processes on multiple clouds, and hence, it is highly impossible for any attacker to re-collect and arrange the keys sequentially for performing decryption. Interestingly, the system only stores the key value but not the location information on run time, which ensures that no intruder could possibly hijack any session. It is also not possible for any technical members of service providers to know the location of secret key. Therefore, the proposed system originally stores the chunks of encrypted splitted data on multiple process of cloud (racks), where it is near to impossible for any hackers to access the complete set of data without any possible information of secret key. This is the pattern by which the proposed system ensures confidentiality, privacy, integrity, as well as non-anonymity of the data.

VII. Result Discussion

The outcome of the proposed system (S^3-IDC) has been evaluated for its effectiveness with respect to transmission rate, encryption time, decryption time, and delay. All the outcomes were observed for test-data size of 1-10 GB. For better effectiveness of the study, the outcomes were also observed for conventional encryption algorithms (SHA1/2 and AES)

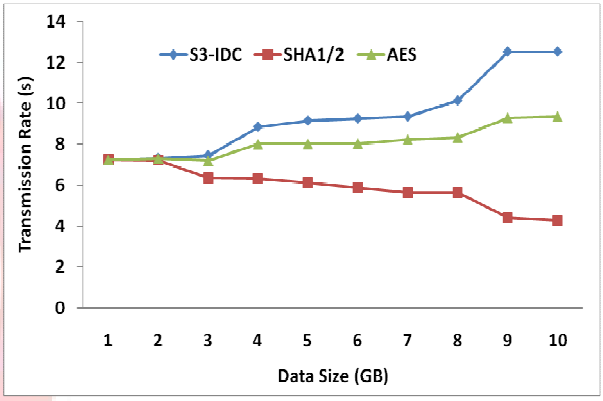

Figure 8 Analysis of Transmission Rate

The transmission rate were observed for test-data size of 1-10GB using network protocol analyzer called as WireShark [26]. The outcome shows that the proposed system is capable of sustaining the increasing load of the data request of the customers. The curve for SHA1/2 is found with lower peaks as it performs iterative encryption on every data request, whereas AES curve is found with minimal peaks as compared to SHA1/2 owing to less complicated structure with minimal key size. The proposed system adopts the process of splitting the data where the number of generated secret keys are highly dependent on the splitted data. Hence, the rate of transmission for proposed system can process higher datasets in shortest ranges of time.

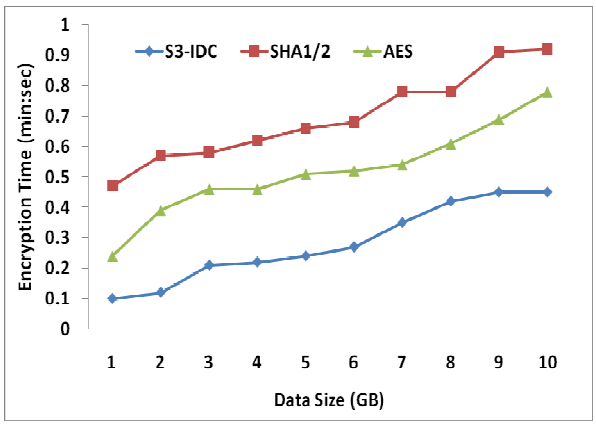

Figure 9 Analysis of Encryption Time

Fig.9 highlights the analysis of encryption time with respect to the increasing data size. It is quite obvious that after performing encryption, the size of the encrypted data do increase in size to some extent which also affects an encryption time. The outcome shows that the encryption time for conventional scheme e.g. SHA1/2 and AES is quite higher compared to proposed S^3-IDC scheme.

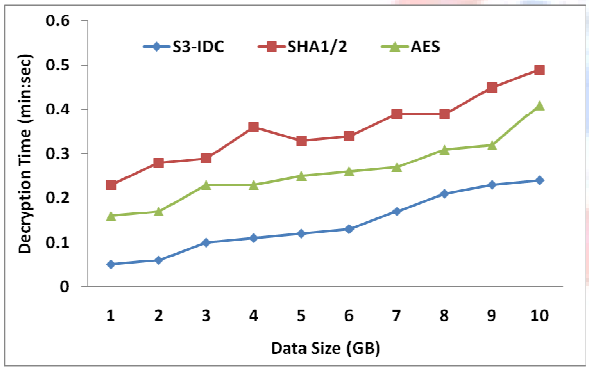

Figure 10 Analysis of Decryption Time

Fig.10 highlights the decryption time of the proposed system along with the conventional system. Decryption time is required to be analyzed for measuring the effectiveness of downloading the encrypted file from the cloud storage. It is also required to understand the success rate in accessibility of the uploaded file from the rack servers. A closer look into the curve will shows that decryption time is approximately reduced compared to the encryption time for almost all the schemes. However, proposed S^3-IDC scheme is witnessed with reduced decryption time, which is also in aggrement with the transmission rate that is increasing order.

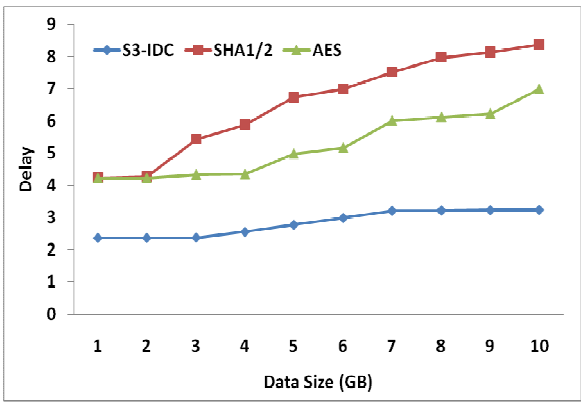

Figure 11 Analysis of Delay

Fig.11 shows the transmission delay in case of large number of data splits on multiple data centers. It could be expected that more the splitting process of customer's data takes place on multiple clouds; more delay could be expected in the peak hours of operation over cloud environment. However, the outcome shows that SHA1/2 as well as AES algorithm is found with increasing trend of delay curve, which is due to increasing steps of encryption and decryption during the request or response of the customers. However, the proposed S^3-IDC scheme performs lightweight cryptography for which reason the decryption time is found to be quite lower as compared to encryption time. Moreover, the adoption of Synchronize Servers and autonomous administrators assists in proper indexing mechanism of the massive chunks of data along with evaluation of the availability of the rack servers. Therefore, although the proposed system exhibit increasing delay with increasing data size, but still it outperforms the conventional security protocols e.g. SHA1/2 and AES, which are currently being used by datacenters for storage security.

VIII. CONCLUSION

The paper mainly concentrates on the data security issues when the customer pays for their storage services on cloud. Although cloud computing offers much cost effective solution towards the data management, but still there are lots of questions pertaining to security loopholes that are constantly being under attention of research community since past decade. After reviewing the research papers that are closely associated with data security on storage applications on cloud, we find that majority of the studies have following pitfalls e.g. i) usage of complex and sophisticated cryptographic technique, where the outcomes doesn't show much conformance to real-time requirements, ii) usage of technique (e.g. deduplication) that has less focus on an effective key management strategies, iii) less emphasis on key management techniques and not much extended analysis has being performed to check sustainability of considered technique. This paper has discussed about a technique that perform a typical encryption on the data that are uploaded by the priviledge customers. Majority of the existing studies were focused on storing the data in one data centers, whereas we choose to design a technique that splits the

requested data of one customer, generates multiple secret key, encrypt those keys, and stores the keys randomly in available racks. Our scheme doesn't entitle any actor to have a possession of the secret key or have any priviledge to access the location information of the data or the secret key. This will mean that a data of single user is converted to multiple chunks of encrypted data and stored on multiple data center randomly. Hence, it is near to impossible for any attacker to have a possession of the data at any cost. If the attacker have a possession of any chunk of encrypted data, it is of no use as the attacker will never have any information of location of remaining chunks of data. Moreover in order to perform decryption, an attacker will have dependency of the key, which is again impossible for them to identify the storage location of the key. Hence, our system will highly discourage any attacker in any process of sniffing the data in cloud storage system. We have also performed comparing our technique using existing SHA1/2 and AES algorithm to find our technique furnishes much better security performance without any significant impact on the data communication over cloud.

REFERENCES

[1] L. Grandinetti, O. Pisacane, M. Sheikhalishahi, Pervasive Cloud Computing Technologies: Future Outlooks and Interdisciplinary Perspectives, IGI Publication, Advances in Systems Analysis, Software Engineering, and High Performance Computing, ISBN-13: 978-1466646834, 2013

[2] G.R. Vijay, A.R.M. Reddy, "Investigational Analysis of Security Measures Effectiveness in Cloud Computing: A Study", Computer Engineering and Intelligent Systems, Vol.5, No.7, 2014

[3] P. Mell, T. Grance, The NIST Definition of Cloud Computing, Recommendations of the National Institute of Standards and Technology, Special Publication 800-145, 2011

[4] A. J. Adoga, G. M. Rabiu, A. A. Audu, "Criteria for Choosing An Effective Cloud Storage Provider", International Journal of Computational Engineering Research, Vol.04, Iss.2, 2014

[5] R.A. Popa., J.R. Lorch., D. Molnar., H.J. Wang., and L. Zhuang., "Enabling Security in Cloud Storage SLAs with Cloud Proof", In USENIX Annual Technical Conference, Vol. 242, 2011

[6] Y. Tang., P.P.C Lee., J.C.S Lui., and R. Perlman., "FADE: Secure overlay cloud storage with file assured deletion", In Security and Privacy in Communication Networks, Springer Berlin Heidelberg, pp. 380-397. 2010

[7] W. Ren., L. Yu., R. Gao., F. Xiong., "Lightweight and Compromise Resilient Storage Outsourcing with Distributed Secure Accessibility in Mobile Cloud Computing", TSINGHUA Science and Technology, Vol. 16, No. 5, pp. 520-528, 2011

[8] X. Dong., R. Li., H. He., W. Zhou., Z. Xue., and H. Wu., "Secure Sensitive Data Sharing on a Big Data Platform", TSINGHUA Science and Technology, Vol. 20, No. 1, pp. 72-80, 2015

[9] A. Bessani., M. Correia., B. Quaresma., F. Andre, and P. Sousa., "DepSky: dependable and secure storage in a cloud-of-clouds", ACM Transactions on Storage (TOS), Vol. 9, No. 4, 2013

[10] J. Stanek., A. Sorniotti., E. Androulaki., and L. Kencl., "A secure data deduplication scheme for cloud storage", In Financial Cryptography and Data Security Springer Berlin Heidelberg, pp. 99-118, 2014

[11] B.H. Kim., W. Huang., and D. Lie., "Unity: secure and durable personal cloud storage", In Proceedings of the 2012 ACM Workshop on Cloud computing security workshop, pp. 31-36, 2012

[12] N. Cao., S. Yu., Z. Yang., W. Lou., and Y. T. Hou., "Lt codes-based secure and reliable cloud storage service", In INFOCOM, Proceedings IEEE, pp. 693-701, 2012

[13] S.Murthy., "Cryptographic Secure Cloud Storage Model with Anonymous Authentication and Automatic File Recovery", ICTACT Journal on Soft Computing, Vol. 5, No. 1, 2014

[14] H. Xiong., X. Zhang., D. Yao., X. Wu., and Y. Wen., "Towards end-to-end secure content storage and delivery with public cloud", In Proceedings of the second ACM conference on Data and Application Security and Privacy, pp. 257-266, 2012

[15] P. Gasti., G. Ateniese., and M. Blanton., "Deniable cloud storage: sharing files via public-key deniability", In Proceedings of the 9th annual ACM workshop on Privacy in the electronic society, ACM, pp. 31-42, 2010

[16] S. Kamara., C. Papamanthou., and T.Roeder., "Cs2: A searchable cryptographic cloud storage system", Microsoft Research, TechReport MSR-TR, Vol.58, 2011

[17] N. Kaaniche., A. Boudguiga., and M. Laurent., "ID-Based Cryptography for Secure Cloud Data Storage", In IEEE Sixth International Conference on Cloud Computing, 2013

[18] S.D.C.D. Vimercati., S. Foresti., S. Jajodia., S. Paraboschi., G. Pelosi., and P.Samarati., "Encryption-based policy enforcement for cloud storage", In Distributed Computing Systems Workshops (ICDCSW), IEEE 30th International Conference, pp. 42-51, 2010

[19] J. Xu., E-C.Chang, and J. Zhou., "Weak leakage-resilient client-side deduplication of encrypted data in cloud storage", In Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security, ACM, pp. 195-206, 2013

[20] P. Puzio., R. Molva., M. Önen., and S. Loureiro., "Block-level de-duplication with encrypted data", Open Journal of Cloud Computing (OJCC), Vol.1, No. 1, pp.10-18, 2014

[21] D.Thilakanathan., S. Chen., S. Nepal., and R. A. Calvo., "Secure Data Sharing in the Cloud", Department of Electrical Engineering, the University of Sydney, 2006

[22] M.K.Margret, "Secure Policy Based Data Sharing for Dynamic Groups in the Cloud", International Journal of Advanced Research in Computer Engineering and Technology (IJARCET), Vol. 2, Issue. 6, 2013

[23] I. Damgård., T.P. Jakobsen., J. B.Nielsen., and J. I. Pagter., "Secure key management in the cloud", In Cryptography and Coding, Springer Berlin Heidelberg, pp. 270-289, 2013

[24] P.K. Tysowski., and M. A. Hasan, "Re-Encryption-Based Key Management towards Secure and Scalable Mobile Applications in Clouds", IACR Cryptology e-Print Archive, Vol. 668, 2011

[25] https://owncloud.com/challenges-secure-cloud-storage/

[26] https://www.wireshark.org/