IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

Privacy Concealment and Data Security over Outsourced Cloud Environments

Sunit Ranjan Poddar[1], Ameya Ravindra Sawant[2], Diksha Pawankumar Jangid[3], Sanika Sarang Kamtekar[4], Prof. Mandar Mokashi[5]

Dept. of Computer Engineering, Dr. D. Y. Patil, Lohegaon, Pune, Maharashtra, India [1,2,3,4,5]

Abstract: In an era of wireless internet, the use of mobile devices and cellphones has reached to a great extent. In this new digital world the availability of data accessibility over internet plays an important role. Dynamic storage systems are preferred for storing huge amount of diverse data. The cloud servers are capable of processing and storing any amount of data required, furthermore the cloud servers are flexible and scalable to great extent. The information on these servers is of huge importance to the data owners and thus require protective strategies. We thus propose a system which consists a group of techniques resulting in providing a secure data storage and data integrity. We have used AES algorithm for encrypting and decrypting the data and digital signatures. We have also used SHA1 for generating hash keys. These hash keys help us support our motive for redundancy checks as well as for maintaining data integrity. Our system also equip de-duplication of user data which promises efficient bandwidth utilization without harming the privacy of the user. Load balancing is another feature that we include in our system for better performance and storage measures. We have used drop boxes to implement our load balancing feature. We strongly believe that the existing systems do not provide the features included in our proposed framework.

Keywords: Data Integrity, Load Balancing, Deduplication, Third Party Auditor.

I. INTRODUCTION

Cloud computing has acquired many interests and accomplishments over the past decade. Cloud services form an efficient platform for many commercial organizations to expand and enhance their business. 'Cloud' deals with a highly variant and huge amount of data. Management and security of such data are considered very tedious tasks. Furthermore, data redundancy is another issue which may cause problems for the cloud services in future. To address all these probable obstacles in cloud computing services, we have implemented a framework which give exceedingly great result and enhanced functionality performances. Our aim towards this framework was towards enabling a system of cloud services which can efficiently manage the data storage over the cloud space along with an ability to detect any kind of tamper done to the client data. A user/client may or may not always access a file once he/she has uploaded it on the cloud space. This calls for a data integrity check over a regular period of time. To achieve these goals, we have implemented the SHA-1 algorithm (Secure Hash Algorithm) for hash key generation and AES (Advanced Encryption Standard) for encryption of client data.

A cloud is a very sophisticated framework which must be accessed and managed with caution. Most of the data stored is in the form of various files. Considering this, our framework divides the user files into number of chunks according to the size of file and stores it over different locations on the cloud. Thus avoiding a chance of gathering the data in only one corner of the cloud. By doing this, security factors are enhanced as an intruder may not have much of a choice if one isn't able to merge the data.

A Third Party Auditor is appointed to look over the functionality of Data Integrity Check from time to time. The TPA notifies the client that his/her data has been altered if TPA finds any kind of change in the hash value of the desired files.

The encryption of the client data is done whenever a new file is being uploaded to the server. A block cipher encryption method has been used in our framework. AES is highly complex and difficult to penetrate through as the data goes through many stages of encryption. Decryption of the data through AES is simply the reversion of the encryption steps. To avoid the loss of important data, we have implemented a data backup feature which stores the files on another location for security purposes. It has been taken care of that the de-duplication factor remains 2.

In all, our framework has shown some excellent results and can comparatively perform better than the existing system is our belief.

II. RELATED WORK

Cloud computing is a boon in today's world where we have to deal with such tremendous amount of data. But along with this, many security concerns such as data security, application security, network security, etc[1][2] have aroused. Data stored in cloud may deal with different types of threats such as data integrity[3]. Data integrity[3][5] refers to assuring and maintaining the consistency and accuracy of data over its entire life cycle over the cloud. A Third Party Auditor[6][9] is used for ensuring the correctness of the data. As used by [22], TPA is used as an automated system which is responsible for

monitoring confidentiality and integrity of the data. Along with maintaining integrity of data, another probable threat that concerns is the storage security. [23] have used homomorphic tokens with distributed verification of erasure-coded data to ensure that unauthorized access is prohibited in cloud network whereas [25] proposes to uses implicit storage security to data and identification based authentication using IBS schemes.

Distributing the load on cloud provides both security and edge in performance measures. This is called Load balancing on the cloud network. [11][12][30][31] brought us the concept of load balancing feature in cloud framework. [12] uses DHT (distributed hash table) based load balancing schema and ALG (application-aware local-global source) for deduplication schema. Along with load balancing, de-duplication is again an important factor in cloud computing. De-duplication means that any replicas of same files are discarded in order to save valuable space. Deduplication is a specialized data compression practice for eliminating duplicate copies of the repeating data. As proposed by [1],[11] and [33], deduplication will give significant strength to cloud in a way that valued space is saved, also this gives performance edge in cloud computing as it does not have to perform same tasks on same files again and again. This de-duplication is achieved by using hash tables[1][11][12][20] which are used to verify if the new file already exists on cloud.[1][15] uses SHA algorithm to generate hash keys. Also, to provide some more security to data on cloud, encryption of the file is needed. Using encryption techniques such as AES[1][17], provides additional security to the cloud. Also, AES is better than DES in performance measures and in someway, security concerns also as AES generates a longer key as compared to DES algorithm.

III.IMPLEMENTED SYSTEM

The entire framework works on two major modules. The first module consists of a user/client who wants to store and process his data in a way that he can save large amount of data in a place which is both secured and gives fast access and processing power. Cloud server helps us to complete our motive. We're using Amazon Web Services (AWS) as a cloud server which is providing us processing and management of data. AWS is also providing us with mailing services that we'll need later which will serve as verification and authorization of users.

At first, the user registers himself on the server to store his data and utilize the services provided by cloud. For the purpose of verification and authorization of the user, we're using Simple e-Mail Service (SeS). SeS provides us with SMTP protocol for mailing purpose, which is followed by a verification stage. Along with verification, SeS also distributes unique download key per registered user which is computed by the cloud server.

Once the user is logged in, he can work on the cloud i.e. upload his file and download any desired files provided he has genuine download key. While uploading the file, cloud server acts as a mediator who handles all the processing on the data before it is uploaded to the cloud space. To serve the purpose of de-duplication, we need such a system which can check whether there was any other file which was uploaded and has the exact same data available. In such a case we don't need to upload this file again as this will only increase redundancy. As an alternative to this, a pointer to previously uploaded file can be placed but with the name that the user desires for his file. In this way, our purpose of de-duplication is fulfilled and the user has his file uploaded on cloud space as he desired. This purpose is contended by using some hashing algorithm. A hash table is created in which all the generated hash values are saved and later used to check if any other file has the same hash value. SHA-1 aids this purpose in our framework. A hash value is generated using the byte code of the file. This hash value is checked against all the previously saved hash values present in the hash table. If we get a hit with exact same hash value, it means that the file is already uploaded on the cloud space which gives us the privilege to discard this file and create a pointer to previously uploaded file. Of course the new file will be of the name that the user desires but it will save significant amount of space when we're working on such tremendous amount of data. Also, this hash value will serve us for checking the data integrity.

It is possible that some of the files may get tampered while uploading or downloading. To inhibit our system through such incidents a system is required which can either detect or stop such anomalies. Our system uses a third party auditor (TPA) to serve this purpose. Using the same hash table generated previously it checks whether any file's hash key is changed. If the respective file's hash value differs from the original hash value, this means that the file has been tampered. In such a case the user is notified and provided with a replica of the original file.

File as a plain text is not as safe as it is when it is encrypted. Such a file with plaintext data is prone to several threats such as data tamper, repudiation and many more. Encryption serves as a countermeasure to such threats. Advanced Encryption Standard (AES) assists our system for this purpose. As the name suggests, AES is an encryption algorithm which uses several rounds in which each round consists of several stages and the data block is transformed from one stage to another in order to create cipher text. The features provided by AES enables us to encrypt the data with a comparatively larger key length. This key is further required to decrypt the file. Along with the security that AES provides, it is resistant to most of the known attacks in today's world. Also, AES works at multiple network layers simultaneously which in turn enables us modify our system in the later stages.

Once the file is encrypted, it needs to be saved on the cloud space. It is always better to divide the file into chunks and save them at different locations rather than keeping the whole file at the same site. This servers as a method by which threats such as file tamper can be avoided. Even if one chunk is tampered, it would still be difficult to tamper all the chunks of the file. For the exact same reason we're dividing the file into chunks. For the ease of computations, these chunks are of similar sizes based on the total size of the file. These chunks are then

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

distributed over cloud in such a way that one chunk is saved on one cloud space from the network. In this way another feature of our framework is achieved and the load is distributed over a network of cloud spaces. Also, this gives us an edge towards security of the file.

AWS also provides us with two more features viz. Relational Database Services (RDS) and Elastic Beanstalk. RDS manages the databases that the system is using apart from cloud space. As we have seen that all the hash values needs to be saved for later references in a hash table, the hash table hence needs to be saved. RDS provides us with a scalable relational database in the cloud where all such data is saved.

Elastic beanstalk is another feature provided by AWS which is used to deploy the framework and manage it without distressing about the infrastructure that runs such applications. It also reduces the management complexities without restricting our choice and control which makes it the perfect candidate for our framework. Elastic Beanstalk launches the suitable environment and creates and configures the required AWS resources to run the code automatically.

A. Algorithms Used

A. SHA-1

SHA algorithm stands for Secure Hash Algorithm used for encryption and decryption of a message. It was designed by NSA in United States of America (1993). Although there have been advances in SHA Encryptions, for e.g., SHA 2, SHA 256; we have chosen SHA 1 implementation for our project. SHA 1 creates a 160-bit message, wherein it goes through the process of padding and appending bits to make it into an even multiple of 512 bits. We have used 80 processing functions defined as follows:

- $f(t;X_1, X_2,X_3) = (X_1 \text{ AND } X_2) \text{ OR } ((\text{NOT } X_1) \text{ AND } X_3)$ **[0 <= t <= 19]**
- $f(t;X_1, X_2, X_3) = X_1 \text{ XOR } X_2 \text{ XOR } X_3$ **[20 <= t <= 39]**
- $f(t;X_1, X_2, X_3) = (X_1 \text{ AND } X_2) \text{ OR } (X_1 \text{ AND } X_3) \text{ OR } (X_2 \text{ AND } X_3)$ **[40 <= t <=59]**
- $f(t;X_1, X_2, X_3) = X_1 \text{ XOR } X_2 \text{ XOR } X_3$ **[60 <= t <= 79]**

The main task of SHA1 algorithm is to loop through the padded and appended message in 512-bit blocks.

- M[1, 2, ..., L]: Blocks of the padded and appended message
- f(0; X1 , X2, X3), f(1, X1, X2, X3), ..., f(79, X1, X2, X3): 80 Processing Functions
- K(0), K(1), ..., K(79): 80 Processing Constant Words
- H0, H1, H2, H3, H4, H5: 5 Word buffers with initial values

For loop on k = 1 to L

- (W(0),W(1),...,W(15)) = M[k] /* Divide M[k] into 16 words */

For t = 16 to 79 do:

- W(t) = (W(t-3) XOR W(t-8) XOR W(t-14) XOR W(t-16)) <<< 1
- X0 = H0, X1 = H1, X2 = H2, X3 = H3, X4 = H4

For t = 0 to 79 do:

- TEMP = X0<<<5 + f(t; X1 ,X2, X3) + X4 + W(t) + K(t) X4 = X3, X3 = X2,
- X2 = X1<<<30, X1 = X0, X0 = TEMP

End of for loop

- H0 = H0 + X0, H1 = H1 + X1, H2 = H2 + X2, H3 = H3 + X3, H4 = H4 + X4

End of for loop

B. AES

AES stands for Advanced Encryption Standard. It consists of linked operations like shuffling bits and replacing some inputs with specific outputs. AES performs the operations on the data formulated in bytes and not in bits. Therefore, it treats the 128 bits of a plaintext block as 16 bytes. AES is a symmetric key symmetric block cipher. The number of encryption rounds in AES are variant depending on the length of the key.

Steps for Encryption:

- Each round consists of 4 operations. The first round is given as:
- Add the 128-bit cipher key to the plaintext to develop an AddRoundkey.
- The 16 bytes of input are substituted using the S-box table designed beforehand which gives a matrix of 4 rows and 4 columns.
- Shift each of the four to the left. Shift is carried out as follows:
  - First row is not shifted.
  - Second row is shifted 1 byte to the left.
  - Third row is shifted 2 bytes to the left.
  - Fourth row is shifted 3 bytes to the left.
- This results in giving us a new matrix consisting of the shifted 16 bytes with respect to each other.
- Each column of the 4 bytes is transformed using mathematical and matrix multiplication functions. A column is replaced by the product of matrix multiplication operation which results into handing us completely new bytes. (This step not performed in last round).

$$\begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

- The above generated matrix is XORed with the 128-bit Round key.
  - If it's the last round, then cipher text is generated.
  - Otherwise another similar encryption round is carried on.

Steps for Decryption

The decryption process is similar to the encryption process in reverse order. Even the sub-processes are implemented in reverse manner.

B. System Algorithm

1. User registration and login

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

2. Verification and distribution of unique download key to each registered user via mail
3. User uploads the file to cloud server

a. Byte code is generated of respective file which helps in creating Byte-array.
b. This byte array is given to our hashing algorithm viz. SHA-1 which in turn generated the hash value for a file
c. This hash value is checked against previously generated hash values of other file from hash table.

i. If the same hash value is present in the table, it means that the file already exists on cloud. In such a case the new file is pointed towards the file with same hash value. In this way, upload of same file with exact data is avoided hence de-duplication is achieved.
ii. If we do not find the same hash value in the table, a new instance of the file is created.

d. File encryption.
d. File is divided into chunks based on file size and distributed amongst network of cloud space

4. While downloading the file

iii. The file is retrieved in the cloud server and merged back
iv. The file is decrypted and given to the user.

5. For data integrity, TPA checks the hash value of the file with the original hash value. If the hash values are changed, the user is notified and if instructed the file is replaced with the original file

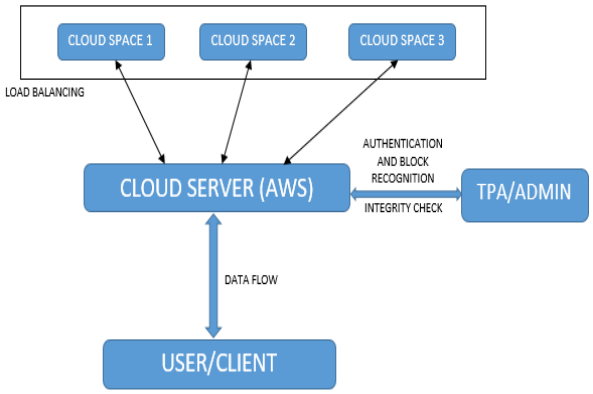

Fig: System Architecture

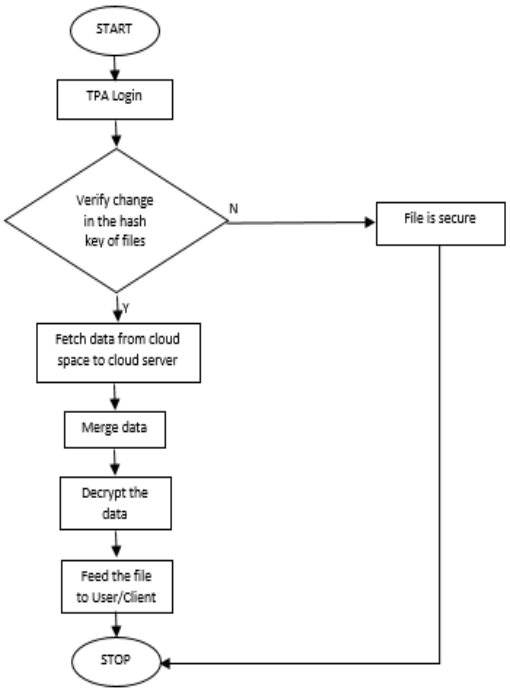

Fig: Flow Chart – Data Integrity

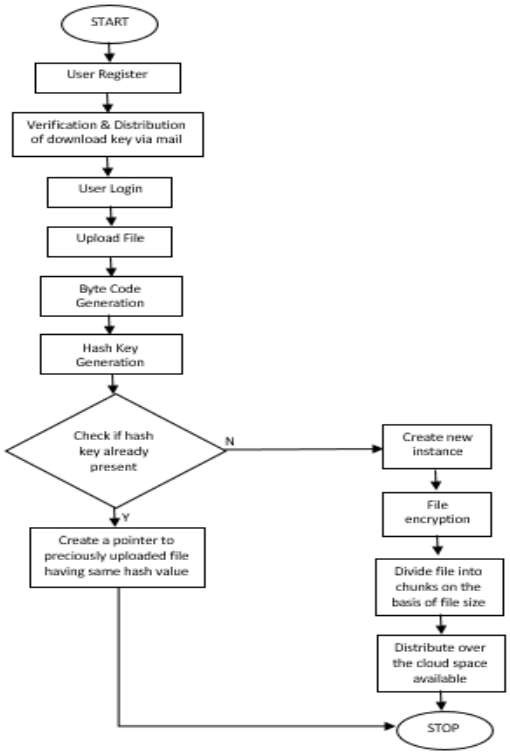

Fig: Flow Chart – File Upload

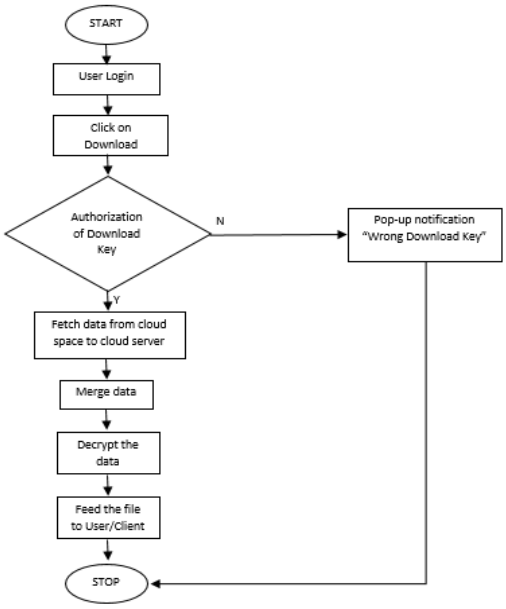

Fig: Flow Chart – File Download

IV. CONCLUSION

The Cloud Security is a very tedious issue which brought our attention towards developing this framework. We have been partially successful in implementing a system which provides complete data integrity and maintaining privacy of the user. Our framework also enhances the Cloud service performances due to the load balancing factor implemented in it. Data de-duplication is prohibited on a great scale which enhances storage management issues. Although many improvements can be made over our framework, it is an honest attempt to make a contribution towards the field.

ACKNOWLEDGMENT

We would like to thank Dr. D. Y. Patil School of Engineering for providing us with all the required amenities. We are also grateful to **Prof. S. S. Das,** Head of Computer Engineering Department, DYPSOE, Lohegaon, Pune for their indispensable support, suggestions and motivation during the entire course of the project.

REFERENCES

[1] Sunit Ranjan Poddar, Sanika Sarang Kamtekar, Ameya Ravindra Sawant and Diksha Jangid, "Privacy Concealment and Data Security over Outsourced Cloud Environments", IJCTA | Nov-Dec 2015 Vol 6 (6), 961-965, ISSN: 2229-6093

[2] Kevin Hamlen, Murat Kantarcioglu, Latifur Khan and Bhavani Thuraisingham, "Security Issues for Cloud Computing", Technical report UTDCS-02-10, Dept of Computer Science, The University of Texas, Dallas, February 2010

[3] Rabi Prasad Padhy, Manas Ranjan Patra and Suresh Chandra Sarapathy, "Cloud Computing: Security Issues and Research challenges", IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS) Vol. 1, No. 2, December 2011

[4] Jian Liu, Kun Huang, Hong Rong, Huimei Wang, and Ming Xian, "Privacy-Preserving Public Auditing for Regenerating-Code-Based Cloud Storage", IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 10, NO. 7, JULY 2015

[5] Henry C.H. Chen and Patrick P.C. Lee, "Enabling Data Integrity Protection in Regenerating-Coding-Based Cloud Storage: Theory and Implementation", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 25, NO. 2, FEBRUARY 2014

[6] Yogita Gunjal, Prof. J. Rethna and Virjil Jeny, "Data Security and Integrity of Cloud Storage in Cloud Computing", International Journal of Innovative Research in Science, Engineering and Technology, Vol. 2, Issue 4, April 2013, ISSN: 2319-8753

[7] Satyakshma Rawat, Richa Chowdhary and Dr. Abhay Bansal, "Data Integrity of Cloud Data Storages (CDSs) in Cloud", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 3, March 2013, ISSN: 2277 128X

[8] Dr. Nedhal A Al-Saiyd and Nada Sail, "DATA INTEGRITY IN CLOUD COMPUTING SECURITY", Journal of Theoretical and Applied Information Technology, December 2013. Vol. 58 No.3, ISSN: 1992-8645

[9] Cong Wang, Sherman S M Chow, Qian Wang, Kui Ren and Wenjing Lou,"Privacy-Preserving Public Auditing for Secure Cloud Storage"

[10] Anuradha R and Dr. Y Vijayalatha, "A Distributed Storage Integrity Auditing for Secure Cloud", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 8, August 2013 ISSN: 2277 128X

[11] S Rajekar, "DATA DE-DUPLICATION AND LOAD BALANCING ON DISTRIBUTED FILE SYSTEM IN CLOUD", Journal of Science and Innovative Engineering & Technology, ISBN 978-81-904760-7-2

[12] D K Karthika, G Sudhakar and D Sugumar, "Efficient Cloud Storage Management Using DHT Mechanism", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, Issue 3, March 2015, ISSN(Online): 2320-9801

[13] Raluca Ada Popa, Frank H Li and Nickolai Zeldovich," An Ideal Security for Order Preserving Encoding"

[14] Vinaya V and Sumathi P, "Implementation of Effective Third Party Auditing for Data Security in Cloud", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 5, May 2013 ISSN: 2277 128X

[15] Piyush Gupta and Sandeep Kumar, "A Comparative Analysis of SHA and MD5 Algorithm", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, ISSN: 0975-9646

[16] Vaishnavi Moorthy and Dr. S Sivasubramaniam,"Implementing Remote Data Integrity Checking Protocol for Secured Storage Services with Data Dynamics and Public Verifiability In Cloud Computing", IOSR Journal of Engineering, Mar. 2012, Vol. 2(3) pp: 496-500

[17] B Padmavathy and S Ranjitha Kumari,"A Survey on Performance Analysis of DES, AES and RSA Algorithm along with LSB Substitution Technique", International Journal of Science and Research (IJSR), India Online ISSN: 2319-7064

[18] Rupani, Shaan, "Amazon Web Services", Source: Target.com

[19] KD Singh and Leo Zhadanovsky, "Setting Up Multiuser Environments in the AWS Cloud (for Classroom Training and Research)", October 2013

[20] P.B.Mane, N.B. Pokale and R.V.Powar, "Storage Optimization on cloud using De-duplication", International Journal of Innovative Research in Advanced Engineering (IJIRAE), Volume 1 Issue 10 (November 2014), ISSN: 2349-2163

[21] Charmee V. Desai and Prof. Gordhan B. Jethava, "Survey on Data Integrity Checking Techniques in Cloud Data Storage", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 12, December 2014 ISSN: 2277 128X

[22] Miss. Nupoor M. Yawale and Prof. V. B. Gadichha, "Third Party Auditing (TPA) for Data Storage Security in Cloud with RC5 Algorithm", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 11, November 2013 ISSN: 2277 128X

[23] Deepanchakaravarthi Purushothaman and Dr.Sunitha Abburu, "An Approach for Data Storage Security in Cloud Computing", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 2, No 1, March 2012 ISSN (Online): 1694-0814

[24] Cong Wang, Qian Wang, Kui Ren, Ning Cao and Wenjing Lou, "Towards Secure and Dependable Storage Services in Cloud Computing"

[25] V. Spoorthy, M. Mamatha, B. Santhosh Kumar, "A Survey on Data Storage and Security in Cloud Computing", IJCSMC, Vol. 3, Issue. 6, June 2014, pg.306 – 313, ISSN 2320–088X

[26] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica and Matei Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing", Technical Report No. UCB/EECS-2009-28 http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html, February 10, 2009

[27] Li Zhao, Ravi Iyer, Srihari Makineni and Laxmi Bhuyan, "Anatomy and Performance of SSL Processing"

[28] Md. Alam Hossain, Md. Biddut Hossain, Md. Shafin Uddin and Shariar Md. Imtiaz, "Performance Analysis of Different Cryptography Algorithms", International Journal of Advanced Research in Computer Science and software Engineering, Volume 6, Issue 3, March 2016 | ISSN: 2277 128X

[29] Mayanka Katyal and Atul Mishra, "A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment", http://www.publishingindia.com

[30] Soumya Ray and Ajanta De Sarkar, "EXECUTION ANALYSIS OF LOAD BALANCING ALGORITHMS IN CLOUD COMPUTING ENVIRONMENT", International Journal on Cloud Computing: Services and Architecture (IJCCSA), Vol.2, No.5, October 2012

[31] Rajwinder Kaur1 and Pawan Luthra2, "Load Balancing in Cloud Computing", Proc. of Int. Conf. on Recent Trends in Information, Telecommunication and Computing, ITC

[32] "Amazon Web Services: Overview of Security Processes", August 2015, http://aws.amazon.com/security/

[33] B. J. MCKENZIE, R. HARRIES AND T. BELL, "Selecting a Hashing Algorithm"

[34] Alexandr Andoni and Piotr Indyk, "Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions"