Journal of Computer Science 10 (10): 1929-1938, 2014
ISSN: 1549-3636

doi:10.3844/jcssp.2014.1929.1938 Published Online 10 (10) 2014 (http://www.thescipub.com/jcs.toc)

METADATA DRIVEN EFFICIENT KEY GENERATION AND DISTRIBUTION IN CLOUD SECURITY

Anitha, R. and Saswati Mukherjee

Department of Information Science and Technology, Anna University, Chennai, India

Received 2013-12-27; Revised 2014-01-16; Accepted 2014-05-12

ABSTRACT

With rapid development of cloud computing to a greater extent IT industries outsource their sensitive data at cloud data storage location. To keep the stored data confidential against untrusted cloud service providers, a natural way is to store only encrypted data in the cloud severs and providing an efficient access control mechanism using a competent cipher key-C_{mxn}, which is becoming a promising cryptographic solution. In this proposed model the cipher key is generated based on attributes of metadata. The key problems of this approach includes, the generation of cipher key-C_{mxn} and establishing an access control mechanism for the encrypted data using cipher key, where keys cannot be revoked without the involvement of data owner and the Metadata Data Server (MDS), hence makes data owner feels comfortable about the data stored. From this study, we propose a novel Metadata driven efficient key generation and distribution policies for cloud data security system by exploiting the characteristic of the metadata stored. Our design enforces security by providing two novel features. 1. Generation of Cipher key-C_{mxn} using modified feistel network, which holds good for the avalanche effect as each round of the feistel function, depends on the previous round. 2. A novel key distribution policy is designed where the encryption and decryption keys cannot be compromised without the involvement of data owner and the Metadata Data Server (MDS), hence makes data owner comfortable about the data stored. We have implemented a security model that incorporates our ideas and evaluated the performance and scalability of the secured model.

Keywords: Security, Data Storage, Metadata, Cloud, Feistel Function, Steganography

1. INTRODUCTION

Cloud computing has become the most attractive field in industry and in research. The requirement for cloud computing has increased in recent days due to the utilization of the software and the hardware with less investment (Anitha and Mukherjee, 2011). A recent survey regarding the use of cloud services made by IDC, highlights that the security is the greatest challenge for the adoption of cloud computing technology (Kuyoro *et al*., 2011). The four key components of data security in cloud computing are data availability, data integrity, data confidentiality and data traceability. Data traceability means that the data transactions and data communication are genuine and that the parties involved are said to be the authorized persons (Mathew, 2012). Several studies shows that data traceability mechanism have been introduced, ranging from data encryption to intrusion detection or role-based access control, doing a great work in protecting sensitive information. However, the majority of these concepts are centrally controlled by administrators, who are one of the major threats to security (Heurix *et al*., 2012). In some modern distributed file systems, data is stored on devices that can be accessed through the metadata, which is managed separately by one or more specialized metadata servers (Cammert *et al*., 2007). Metadata is a data about data and it is structured information that describes, explains, locates and makes easier to retrieve, use, or manage an information resource. The metadata file holds the information about a file stored

Corresponding Author: Vasudevan, P., Department of Information Science and Technology, Anna University, Chennai, India

in the data servers. In cloud computing, the users will give up their data to the cloud service provider for storage. The data owners in cloud computing environment want to make sure that their data are kept confidential to outsiders, including the cloud service provider which will be the major data security requirement. In the existing system, an authentication is done using cloud user's identity and must be validated by the central authority, called cloud service providers. When the cloud service provider is malicious unauthorized users can also be impersonated. Hence we are facing a major issue with Key-Generation and Key handling problem. When the secret key is generated in a single space the system can be easily attacked. Hence in order to overcome these issues, in the proposed cipher keys are generated using the metadata attributes and key handling mechanism hence there doesn't have any centralized control over the encryption and decryption technique. The specification of deciding the key is based on the metadata attribute in the metadata server as well as the user key. Most of the existing cloud encryption schemes are constructed on the architecture where a single Trusted (TPA) third party authority has the power to secure the secret data stored at the cloud servers. The major drawbacks of the prevailing system is that the data stored is not much secure because the entire security is taken care by a single space. Hence in the proposed system, the key generation and issuing protocol is handled by User, MDS and DS. The model also makes data owner confident about the complete security of the data stored, since the encryption and decryption keys cannot be compromised without the involvement of data owner and the MDS. Based on the above-mentioned analysis, it is needed to propose a secure data-sharing scheme, which simultaneously achieves high performance, full delegation and scalable revocation.

Our contributions can be summarized as follows.

We propose a model to create a cipher key-C_{mxn} based on the attribute of metadata stored using a modified feistel network and support user to access the data in a secured mode.

We also propose a novel security policy which involves the data owner, the MDS and the data server by means of key creation and sharing policies thereby ensuring that the model prevents unauthorized access of data.

The rest of the paper is organized as follows: Section 2 summarizes the related work and the problem statement. Section 3 describes the system architecture model and discusses the detailed design of the system model. Section 4 describes the modified feistel network structure design and issues of the proposed model. Section 5 explains about the data security at the data server location. The performance evaluation based on the prototype implementation is given in section 6 and 7 concludes the study.

2. RELATED WORKS

The Related work discusses about the previous work carried out in the area of cloud security and we have also discussed about how metadata is used in cloud computing environment.

2.1. Metadata in Distributed Storage Systems

Recently a large amount of work is being pursued in data analytics in cloud storage Verma *et al*. (2010) have proposed metadata using Ring file system. In this scheme metadata for a file is stored based on hashing its parent location. Replica is stored in its successor metadata server. Hua *et al*. (2011) have proposed a scalable and adaptive metadata management in ultra large scale file systems. According to Cammert *et al*. (2007) metadata is divided into two types: Static and dynamic metadata. The author has suggested publish-subscribe architecture, enabled a SSPS to provide metadata on demand and handled metadata dependencies successfully. Anitha and Mukherjee (2011) has described that the data retrieval using metadata in cloud environment is less time consuming when compared to retrieving a data directly from the data server. Aziz (2011) has discussed that the data quality is increased by the latest metadata analysis.

2.2. Security Schemes

Modi *et al*. (2013) proposed a survey paper where they discussed about factors affecting cloud computing storage adoption, vulnerabilities and attacks. The authors have also identified relevant solution directives to strengthen security and privacy in the cloud environment. They further discuss about various threats like abusive use of cloud computing, insecure interfaces, data loss and leakage, identity theft and metadata spoofing attack. Kumar and Revati (2012) shows that third party auditor is used to periodically verify the data integrity for the data stored at cloud service provider without retrieving the original data. The security is provided by creating the metadata for the encrypted data. Kaneko *et al*. (2011) have proposed a query based hiding schema Information using a Bloom filter. The query given is processed and the attributes of the query is used for key generation. The generated key is used to hide

confidential information. The author Aguilera *et al*. (2003) has proposed a practical and efficient method for adding security to Network-Attached Disks (NADs). The design specifies a protocol for providing access to the remote block-based devices. The security is provided by means of access control mechanism. Abidin *et al*. (2011) a has discussed that the key generation in the form of matrix. He has solved the problem of non invertible key matrix problem. The computational complexity in term of generating the inverse key matrix is reduced by his mechanism.

2.3. Bloom Filter Schemes

The Bloom filter is a space-efficient probabilistic data structure that supports set membership queries (Kumar and Revati, 2012). The data structure was conceived by Burton H. Bloom in 1970. The structure offers a compact probabilistic way to represent a set that can result in false positives (claiming an element to be part of the set when it was not inserted), but never in false negatives (reporting an inserted element to be absent from the set). This makes Bloom filters useful for many different kinds of tasks that involve lists and sets. The basic operations involve adding elements to the set and querying for element membership in the probabilistic set representation. Kaneko *et al*. (2011) has discuss about the usage of bloom filter in query processing.

2.4. Steganography Security Schemes

Wawge and Rathod (2012) describes that steganography comes from the Greek words Steganos (Covered) and Graptos Writing). The term steganography came into use in 1500‘s after the appearance of Trithemius book on the subject Steganographia. The word steganography technically means covered or hidden writing. The proposed new data hiding scheme by using matrix matching method. On this basis of matching factor of columns, particularbits may be changed such that change in image quality is minimum. Thus the original content is hidden (Kaur *et al*., 2012). Govada *et al*. (2012) in the year 2012 proposed text steganography with multi level shielding where he proposed a method which is capable of performing text steganography that is more reliable and secure when compared to the existing algorithms. The method is a combination of word shifting, text steganography and synonym text steganography. Chowdhury and Manna (2012) has proposed an efficient method of steganography using matrix approach. He has discussed that the goal of steganography is to hide messages inside other ‘harmless’ messages in a way that does not allow any enemy to even detect that there is a second message present. Least Significant Bit (LSB) insertion is a common and simple approach to embed information in a cover object which he has used. The design uses a matrix based steganography which modifies the bit inside the matrix by means of adding and modifying.

3. SYSTEM ARCHITECTURE

The architecture diagram of the proposed system model is shown in **Fig. 1**. Each block in the architecture explains about how the data is encrypted and how the keys are shared between the user, MDS and the DS.

The system model proposes security to the data using modified Feistel network where the metadata attributes are taken as input in the form of matrices. In this model the user uploads the encrypted file using the key X_1. The metadata for the file is created and based on the metadata created, attributes of the cipher key C_{mxn} is created. The Metadata server sends the cipher key C_{mxn} to the user. Using C_{mxn} as key the user encrypts the key X_1 and generates X_2. While downloading the file the key X_2 and C_{mxn} is used to retrieve X_1 and file is decrypted. This model proposes a modified Feistel function F which introduces the matrix operations like transpose, shuffle, addition and multiplication along with the key matrix. The cryptanalysis carried out in this study clearly indicates that this cipher cannot be broken by the brute force attack. This model provides high strength to the cipher, as the encryption key induces a significant amount of matrix obfuscation into the cipher. The avalanche effect discussed shows the strength of the cipher C_{mxn}. The secured bloom filter indexing is used to generate the stego data in order to prevent the data at the data server location. The data stego is generated using the bloom filter index value and the user key K. Thus the original data is made secured at the data server location. The proposed system model also ensures that the data is identically maintained by making use of the cipher key C during any operation like transfer, storage, or retrieval:

- File uploading
- Data pre processing
- Construction of modified feistel network
- Generation of Cipher key C_{mxn}
- Generation of Secured Bloom filter index
- Creation of data stego

The file uploading process is explained in **Fig. 2.**

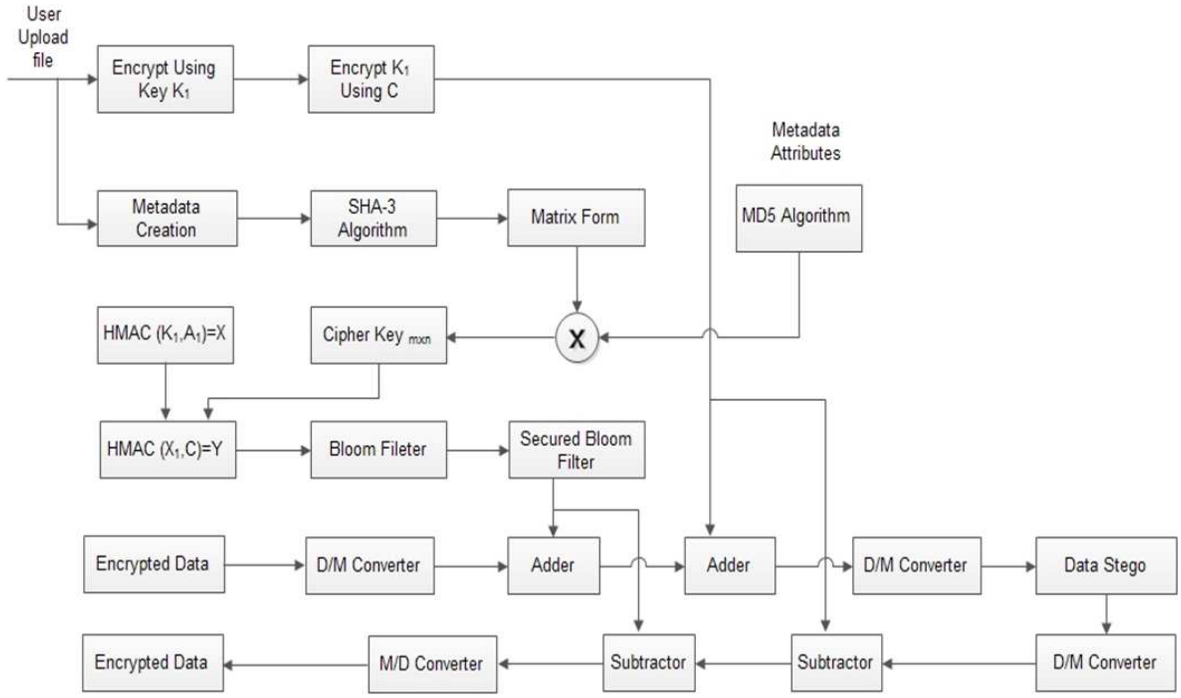

Fig. 1. Architecture diagram

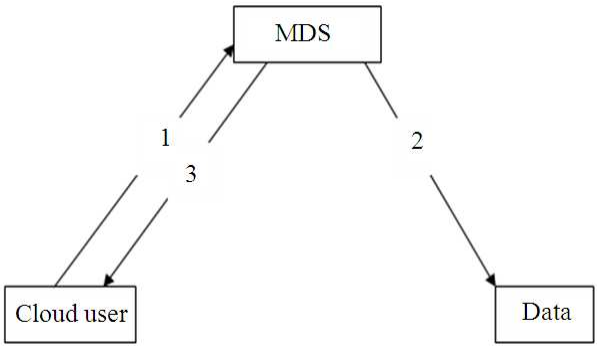

Fig. 2. File upload and download using security key 1. Uploads the encrypted file 2. Encrypted file sent to the data server 3. Sends the Cipher key Cmxn to the user

File Access: When a user sends request for data stored on the cloud environment, the request is given to the metadata server which provides the recent cipher key C_{mxn} to the user. Using Cmxn user decrypts the key X_2 and gets X_1. Using X_1 the encrypted file from the cloud storage is decrypted to get the original data. By providing the recent cipher key C_{mxn} the data integrity is also verified. Our system methodology uses the functionalities:

4. MODIFIED FEISTEL NETWORK

Feistel ciphers are a special class of iterated block ciphers where the cipher text is calculated from the attributes of metadata by repeated application of the same transformation or round function.

4.1. Development of the Cipher Key "C_{mxn}" Using Modified Feistel Function

In this study we propose a complex procedure for generating the cipher key "Cmxn" based on matrix manipulations, which could be introduced in symmetric ciphers. The proposed cipher key generation model offers two advantages. First, the procedure is simple to implement and has complexity in determining the key through crypt analysis. Secondly, the procedure produces a strong avalanche effect making many values in the output block of a cipher to undergo changes with one value change in the secret key. As a case study, matrix based cipher key generation procedure has been introduced in this cloud security model and key avalanche have been observed. Thus the cloud security model is improved by providing a novel mechanism using modified Feistel network where the cipher key C_{mxn} is generated with the matrix based cipher key generation procedure.

4.2. Procedure for Generating Cipher Key C_{mxn}

The Cipher key generation procedure is based on a matrix initialized using secret key and the modified feistel function F. The input values used in various feistel rounds are taken from the previous round. The selection of rows and columns for the creation of matrix is based on the number of attributes of the metadata and the secret key matrix " K_{mxn}" and the other functional logic as explained in the following subsections.

4.3. Data Preprocessing

Data preprocessing is a model for converting the metadata attributes into matrix form using the SHA-3 cryptographic algorithm, containing m rows and n columns, where m is the number of attributes of the metadata and n takes the size of the SHA-3 output. **Figure 3** explains about the data preprocessing algorithm. The matrix is splitted into 4 equal matrix say m_1, m_2, m_3 and m_4. The matrix obfuscation is carried out in order to make the hacker opaque. The matrices m_1, m_3 and m_2, m_4 are concatenated. This obfuscated matrix is fed as input to the feistel network structure where concatenated value of m_1, m_3 will be the left value and m_2, m_4 be the right value of the feistel network.

4.4. Modified Feistel Network Structure

The Matrix L_{mxn} which is a concatenated value of $m_1 \parallel m_3$ is considered as the left value of the feistel network structure and Matrix $R_{mxn} = m_2 \parallel m_4$ is considered as the right value of the feistel network structure. Using MD5 cryptographic hash algorithm the key matrix Kmxn is generated whose size is m x n where "m" is the number of attributes of metadata and "n" is the size of the MD5 algorithm. The development of the cipher key in the feistel network is done through the number of rounds until the condition is satisfied. In this symmetric block ciphers, matrix obfuscation operations are performed in multiple rounds using the key matrix and the right side value of the feistel network structure. The function F plays a very important role in deciding the security of block ciphers. The concatenated value of L_{mxn} and R_{mxn} in the last round will be the cipher key C_{mxn}. **Figure 4** below represents the one round modified feistel network structure.

4.5. Definition of Feistel Function F

Let R be a function variable and let K be a hidden random seed, then the function f is defined as, $F(R, K) = F_K(R)$ where F is a modified feistel function.

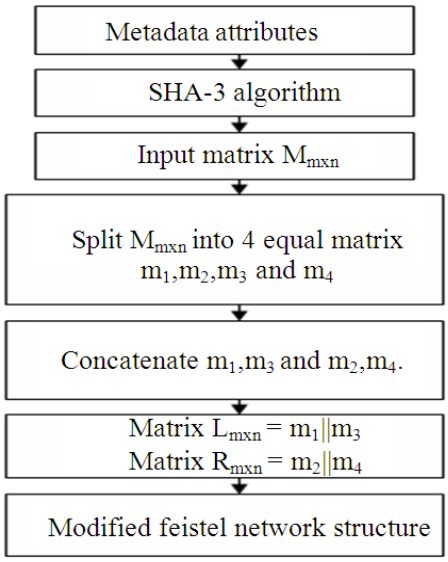

Fig. 3. Model for data preprocessing

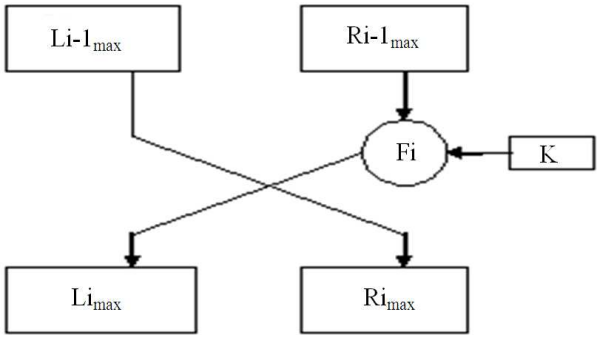

Fig. 4. One round of modified feistel network

The procedure for developing the function is described below. Each round has its own feistel function F. The function f is considered to be varied based on the right side value of the feistel network i.e., the function F is indexed by the matrix R_{mxn} for that round. In this modified feistel network structure, the function for each round depends on the previous round i.e., $Round_i$:

$$F(L_i, R_i) = (R_{i-1}, F(K, L_{i-2}))$$

The above formula shows that a small change in one round affects the entire feistel network. For each round as the value of R of the network gets compressed at some point in time the feistel round automatically stops based on the size of the attributes.

Algorithm 1: Creation of cipher key

Begin
1. Read Metadata attribute
2. Apply SHA-3
3. Generate Matrix $M_{mxn,}$ split the matrix and generate L_{mxn} and R_{mxn}
4. Left value of Feistel = L_{mxn} and Right value of Feistel = R_{mxn}

For i = 1 to n Repeat till n/2 = 1

Begin
4.1 Split R_{mxn} into equal matrix, R_{1mxn}, R_{2mxn}
4.2 Transpose R_{1mxn}, R_{2mxn} as R_{1nxm}, R_{2nxm}
4.3 Apply matrix addition of R_{1nxm}, R_{2nxm} = T_{mxn}
4.4 Transpose T_{mxn}
4.5 Matrix multiplication of $T_{nxm} * K_{mxn} = RV_{mxn}$
/ *condition for multiplication is verified */
4.6 New $L_{mxn} = RV_{mxn}$
4.7 New R_{mxn} = Old value of L_{mxn}
End
5. Repeat the step till n takes odd value
6. Write(C) Cipher key C = L_{mxn} || R_{mxn} / *|| represents concatenation */
End

Algorithm 2: Creation of feistel function F

Begin
1) Read Matrices L_{mxn} and R_{mxn}
a. Assign Left value = L_{mxn}
b. Right value = R_{mxn}
2) Split $R_{mxn} = R_{1mxn}$ and R_{2mxn}.
3) Transpose (R_{1mxn}) = R_{1nxm}
4) Transpose (R_{2mxn}) = R_{2nxm}
5) $T_{nxm} = R_{1nxm} + R_{2nxm}$.
6) R_V mxn. = $T_{nxm} * K_{mxn}$
7) Re - Assign
a. $L_{mxn} = R_{Vmxn}$
b. $R_{mxn} = L_{mxn}$
8 Go to Step 2 till n = odd value.
End

4.6. Analysis of Cipher Key C_{mxn}: Avalanche Effect

The modified feistel network also holds good for the avalanche effect as each round depends on the previous round value. Avalanche effect is an important characteristic for encryption algorithm. This characteristic is seen that a small change in the metadata attribute will have the effect on its cipher key which shows the efficacy of the cipher key i.e., when changing one bit in plaintext and will change the outcome of at least half of the bits in the cipher text. The need for the discussion of avalanche effect is that by changing only one bit in a matrix, leads to a large change in the existing key, hence it is hard to perform an analysis of cipher text, when trying to come up with an attack. The avalanche effect is calculated by the formula:

$$Avalanche\ Effect = \frac{Number\ of\ values\ changed\ in\ the\ cipher\ key\ C_{mxn}}{Total\ number\ of\ values\ in\ the\ cipher\ key\ C_{mxn}}$$

4.7. Generation of Secured Bloom Filter Index

The second level of security in the metadata layer which is provided using the secured bloom filter look up table. The generation of the look up table is as shown in **Fig. 5**.

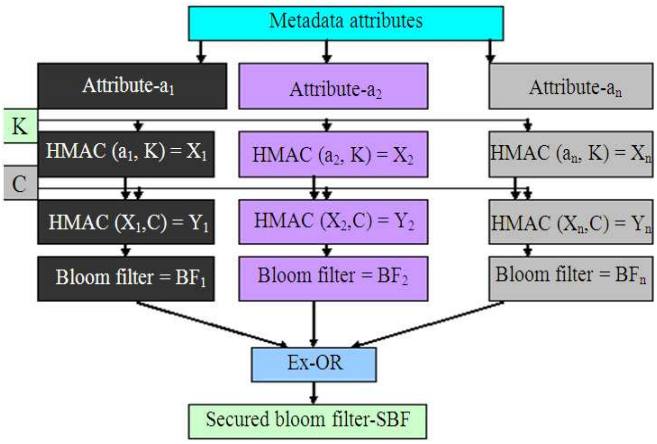

Fig. 5. Secured bloom filter index generation

A secured bloom filter index is created based on the value of cipher key C_{mxn} and key K which is X_2 of the user using the attribute of metadata. The $HMAC_1$ is applied for every attribute A of the metadata created using the key from the user and the output of the first level is again applied for $HMAC_2$ using the cipher key C_{mxn} hence the index creation cannot be compromised without the involvement of the user and the metadata attributes.

5. DATA STEGANOGRAPHY AT DATA SERVER LOCATION

This section explains about the data security at the data server location. As the data in cloud is kept in the data server which is away from the user the security of data at rest plays a major role. The generation of stego data is as shown in the **Fig. 6**. The original data is converted into data stego at the time of storing the data. The conversion process is carried out in 4 steps. 1. Data to Matrix converter 2. Matrix is added with SBF value 3. The output is added with the key from the user 4. Matrix to data convertor. Thus the original data is divided into data stego and uploaded to the data server location. Certain mathematical operations like converting the data block into matrix form and by using SBF, the original matrix is modified. To hide the original information, straight message insertion may transform every bit value of original information i.e., embedding some bit values to the original value. Each of these techniques is applied to provide security to the data, by hiding the original data. For steganography, we have used matrix operations in order to hide the original information. While downloading whatever added to the stego data has to be subtracted and is processed to get original encrypted data as described in the **Fig.7.**

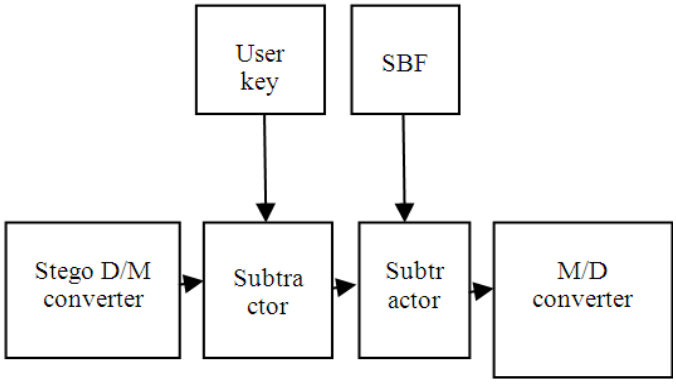

Fig. 6. Generation of stegodata

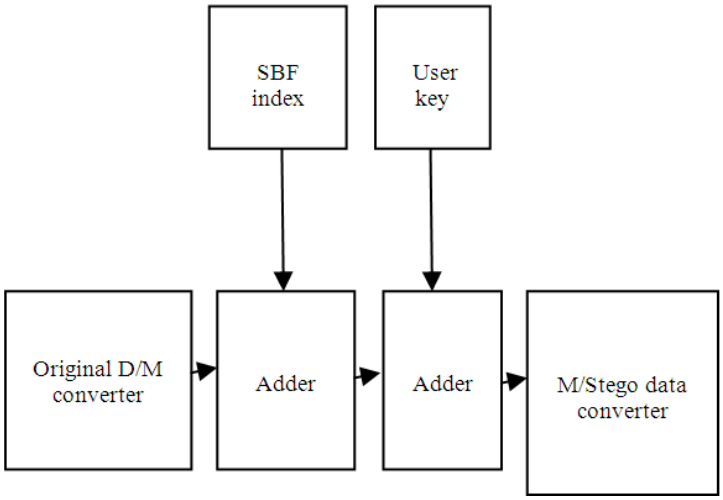

Fig. 7. Conversion of stegodata to original data

6. EXPERIMENTAL RESULTS

The experiments have been carried out in a cloud setup using Eucalyptus which contains cloud controller and walrus as storage controller. These tests were done on 5 node cluster. Each node has two 3.06 GHz Intel (R) Core TM processors, i-7 2600, CPU @ 3.40GHZ, 4 GB of memory and four 512 GB hard disks, running Eucalyptus. The tests used 500 files of real data set, uploaded into the storage and then downloaded based on the user's requirement. The experimental results show that the model provides a complex cipher key C_{mxn} which adequately strengthens the data stored. Results demonstrate that our design is highly complex in nature and the time taken for generating the cipher key is less compared to the existing algorithms. Performance Analysis metrics is done based on the experimental set up. To the best of the domain knowledge obtained due to a wide literature survey on cloud-based performance analysis methodologies and tools, the performance analysis metrics useful for analyzing the cloud security are listed and the comparison results are given. From **Fig. 8** it is observed that the avalanche effect of the proposed system is very high. As the generated key is in matrix format a single change in the bit value will affect the entire key value in the cipher key C_{mxn}. The total time taken for encrption mecanism is as shown in **Fig. 9** with respect to varying file sizes. From the **Fig. 9** it is observed that the total encrption mechanism of the proposed mechanism is less when compared to the existing systems. From the **Fig. 10** it is observed that the total dencryption mechanism of the proposed mechanism is less when compared to the existing systems.

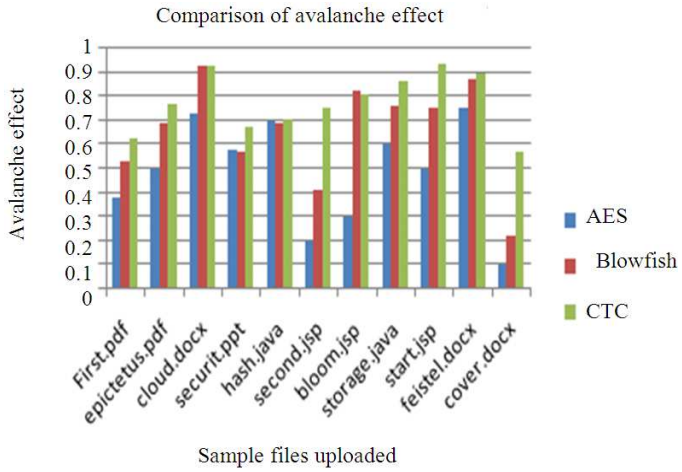

Fig. 8. Comparison of avalache effect of proposed and existing key generation techniques

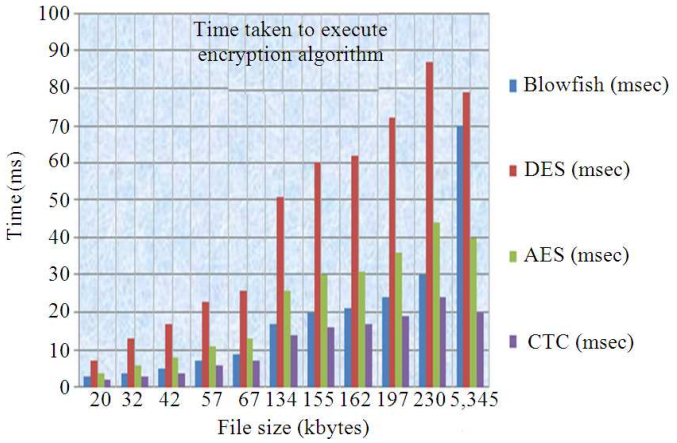

Fig. 9. Comparison of time taken for total Encrption mechanism

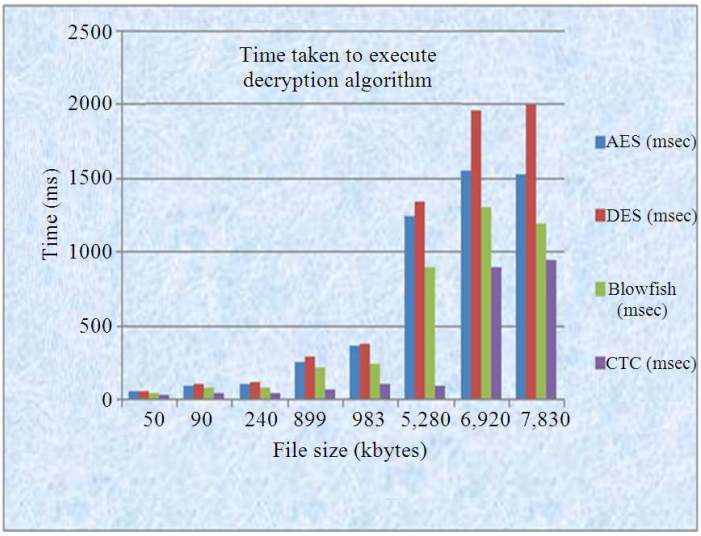

Fig. 10. Comparison of time taken for total Decrption Mechanism

7. CONCLUSION

This study investigates the problem of data security in cloud data storage where the data is stored away from the user. The problem of privacy of data stored has been studied and an efficient and secured protocol is proposed to store data at the cloud storage servers. We believed that the data storage security in cloud era is full of challenges especially when the data is at rest and at the data location. Our method provides privacy to the data stored and the challenge in constructing the security policy, involves both the data owner as well as the MDS to store and retrieve the original data. As the key is in matrix form, it provides major strength to the proposed model. The model also makes data owner confident of the security of the data stored in the centralized cloud environment, since the encryption and decryption keys cannot be compromised without the involvement of both the data owner and the MDS.

8. REFERENCES

Abidin, A.F.A., O.Y. Chuan and M.R.K. Ariffin, 2011. A novel enhancement technique of the hill cipher for effective cryptographic purposes. Am. J. Applied Sci., 7: 785-789. DOI: 10.3844/jcssp.2011.785.789

Aguilera, M.K., M. Lillibridge and J. Maccormick, 2003. Block-level security for network-attached disks. Proceedings of the 2nd Usenix Conference File Storage Technologies, (CST' 03), pp: 159-174.

Anitha, R. and S. Mukherjee, 2011. A dynamic semantic metadata model in cloud computing. Proceedings of the 4th International Conference on Obcom, Dec. 9-11, India, pp: 13-21. DOI: 10.1007/978-3-642-29216-3_3

Aziz, A.A., M.Y.M. Saman and M.P. Hamzah, 2011. Using metadata analysis and base analysis techniques in data qualities framework for data warehouses. Am. J. Econom. Bus. Admin., 3: 112-119. DOI: 10.3844/ajebasp.2011.112.119

Cammert, M., J. Kramer and B. Seeger, 2007. Dynamic metadata management for scalable stream processing systems. Proceedings of the IEEE International Conference on Data Engineering Workshop, Apr. 17-20, IEEE Xplore Press, Istanbul, pp: 644-653. DOI: 10.1109/ICDEW.2007.4401051

Chowdhury, N. and P. Manna, 2012. An efficient method of steganography using matrix approach. Int. J. Intell. Syst. Applic., 4: 32-33. DOI: 10.5815/ijisa.2012.01.04

Govada, S.R. B.S. Kumar, M. Devarakonda and M.J. Stephen, 2012. Text steganography with multi level shielding. Int. J. Comput. Sci., 9: 401-404.

Heurix, J., M. Karlinge and T. Neubauer, 2012. Perimeter-pseudonymization and personal metadata encryption for privacy-preserving searchable documents. Proceedinfs of the 45th Hawaii International Conference on System Sciences, (CSS' 12), Washington, pp: 3011-3020. DOI: 10.1109/HICSS.2012.491

Hua, Y., Y. Zhu, H. Jiang, D. Feng and L. Tian, 2011. Supporting scalable and adaptive metadata management in ultralarge-scale file systems. IEEE Trans. Parellel Distributed Syst., 22: 580-593. DOI: 10.1109/TPDS.2010.116

Kaneko, S. T. Amagasa and C. Watanabe, 2011. Semi-shuffled BF: Performance improvement of a privacy-preserving query method for a daas model using a bloom filter. Proceedings International Conference Parallel Distributed Processing Techniques Applications, (PTA' 11).

Kaur, J., M. Duhan, A. Kumar and R.K. Yadav, 2012. Matrix matching method for secret communication using image steganography. Int. J. Eng.

Kumar, J.R. and M. Revati, 2012. Efficient data storage and security in cloud. Proc. Int. J. Emerg. Trends Eng. Develop.

Kuyoro, S.O, F. Ibikunle and O. Awodele, 2011. Cloud computing security issues and challenges. Int. J. Comput. Netw., 3: 247-255.

Mathew, A., 2012. Survey paper on security and privacy issues in cloud storage systems. Proc. Electr. Eng. Seminar Spec. Probl., 571: 1-13.

Modi, C., D. Patel, B. Borisaniya, A. Patel and M. Rajarajan, 2013. A survey on security issues and solutions at different layers of Cloud computing. J. Comput., 63: 561-592. DOI: 10.1007/s11227-012-0831-5

Tang, Y., P.P.C. Lee, J.C.S. Lui and R. Perlman, 2012. Secure overlay cloud storage with access control and assured deletion. IEEE Trans. Dependable Secure Comput., 9: 903-916. DOI: 10.1109/TDSC.2012.49

Verma, A., S. Venkataraman, M. Caesar and R.H. Campbell, 2010. Efficient metadata management for cloud computing applications. Proceedings of the International Conference Communication Software Networks, (CSN' 10), pp: 514-519.

Wang, Q., C. Wang, K. Ren, W. Lou and J. Li, 2011. Enabling public auditability and data dynamics for storage security in cloud computing. IEEE Trans. Parallel Distrib. Syst., 22: 847-859. DOI: 10.1109/TPDS.2010.183

Wang, Y. and H.T. Lv, 2011. Efficient metadata management in cloud computing.

Wawge, P.U. and A.R. Rathod, 2012. Cloud computing security with steganography and cryptography AES algorithm technology. Proc. World Res. J. Comput. Architecture, 1: 11-15.

Wu, J.J., P. Liu and Y.C. Chung, 2010. Metadata partitioning for large-scale distributed storage systems. Proceedings of the IEEE International Conference Cloud Computing, Jul. 5-10, IEEE Xplore Press, Miami, pp: 212-219. DOI: 10.1109/CLOUD.2010.24

Yu, S., C. Wang, K. Ren and W. Lou, 2010. Achieving secure, scalable and Fine-grained Data Access Control in Cloud Computing. Proceedings of the IEEE Infocom, Mar. 14-19, IEEE Xplore Press, San Diego, pp: 1-9. DOI: 10.1109/INFCOM.2010.5462174