Journal of
Data Acquisition and Processing

ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7747451

SECURED ARCHITECTURE FOR BIG DATA STORAGE IN CLOUD COMPUTING ENVIRONMENT

***A. Vineela**
Research Scholar, Department of CSE, JNTUA, Anantapuramu, Andhra Pradesh, India
Email: vineela177a@gmail.com

N. Kasiviswanath
Professor, Department of CSE, G.Pulla Reddy Engineering College, Kurnool
Andhra Pradesh, India, Email: hodcse@gprec.ac.in

C.Shoba Bindu
Professor, Department of CSE, JNTUA, Anantapuramu, Andhra Pradesh, India
Email: shobhabindhu@gmail.com

Abstract: Despite the increasing popularity of big data, it has been encountering various security issues and storage limitations. Although the combination of Big Data and cloud computing can help improve storage and security, it is still important to have a secure architecture in place to manage it. The goal of this study is to develop a secure architecture that uses the AROMA algorithm for protecting big data. The proposed solution would provide a secure service for storing big data. It would also address the issue of unauthorized access by cloud service providers. The results of an experiment conducted in Amazon EC2 revealed that the proposed method is more efficient.

Keywords: Big Data, Cloud, Security, Encryption, Storage

1. Introduction

Big data [1-3] refers to the set of large volumes of data which is more complex to analyse. Due to the complexity and large volumes of big data, it is difficult to manage those data sets using traditional hardware and software [4]. Big data includes the medical data, machine generated data, sensor data, social media posts, administrative data, blogs, web pages, and military data [5]. Big data storage and maintenance involves high cost due to the requirements of physical resources like hard disks, high potential servers, network resources etc., and it requires continues maintenance procedures like backup and tuning. This will not be possible with available resources with the organizations. The cheaper solution for the above issue is data management in the cloud [6]. Once the data is organized with the cloud service provider, then the third party has the complete authority over the data within their infrastructure with high availability and security.

Cloud provides increased volume of data storage and computing capacity to run users application. It facilitates new ways to the user to access, manage and analyse the big data [7]. The users can access the cloud from anywhere and anytime irrespective of location with the connectivity of internet. Even though, the cloud has more advantages, there are some limitations such as identify management [8], data protection [9], vulnerability management [12] and disaster recovery [13]. These are the major security issues in cloud. Among them, data protection is the most important security issue. Usually security issues are concerned with four factors such as: (1) data input, (2) data and command output, (3) shared policy (4) infrastructure. Each of these factors contains sub-challenges such as data input associates the issues corresponding to data collection and data forwarding corresponding to transportation along with data storage. Therefore, a proper policy is needed to build cloud more secure.

In this paper, a secure architecture is presented for managing big data in cloud. The importance of the proposed approach is to provide security-as-a-service for big data storage. The major issue solved by the proposed approach is restricting the cloud service providers from directly accessing the user's data. The rest of the paper is organized as follows. Section 2 deals with the related work regarding the security approaches in cloud and big data. Section 3 deals with the architecture of big data storage in cloud. Section 4 explains about the Advanced RedistributiOn MetA storage (AROMA) algorithm. Section 5 shows the results analysis of the proposed algorithm. Finally, conclusion is drawn in Section 6.

2. Related work

In the recent years, several researchers are concentrated on developing the security and privacy mechanisms for big data. In [10], the author proposed a new model which is concentrated on user data privacy. The user data is related to the social networking and it is particularly associated with mobile applications. The concept behind this model is explicitly display the user's personal data within the mobile application and managing the security by providing authentication. Though, the mechanism is good in authentication, but failed in protecting the user privacy. In [11], the authors developed a novel framework called as *MyCloud* for solving the security issues in big data through cloud. *MyCloud* restricts the virtual machine controlling software in the client host and assigns trust computing environment that having the performance and security oriented components. This framework provides the flexibility to the users for configuring the security constraints without the virtual machine influence. The computational overhead is the major drawback of *MyCloud*.

Big Data management security is the major aspect in cloud computing, which usually focus on data classification and encryption mechanisms [20, 21]. Some techniques are focused on developing the query processing through Resource Description Framework (RDF). However, a suitable encryption mechanism is needed to ensure in reducing the computing time while processing the big data in cloud [24]. For instance, classification of data with their priorities using searchable encryption algorithm is one of the approaches for users to take the decision whether encryption needs to be performed on the selected data [23]. However, many data management techniques assume that cloud service providers have the limited access to the user's data. In some situations, there is the flexibility to retrieve the data from the cloud even though it is encrypted.

Next issue in the cloud is securing and monitoring the big data in the cloud. It is depend on the examining the cloud operators behaviour. One of the techniques used to secure the data in cloud is Attribute based encryption. However, imposing restrictions on the cloud operators leads to some other issues like data integration and data management [22, 25, 31]. In some situations, blocking of cloud service providers leads to operation failure and data damage. Therefore, an efficient secure architecture is needed to preserve the security and privacy without violating the policies of the cloud service providers [29-30].

In summary, after analysing the existing mechanisms for securing big data in cloud have two solutions. The first solution is to restrict the rights of the employers in cloud service providers. This type of procedure will not be managed by technical methods. Other solution is to protect the big data by employing the encryption techniques such as attribute based encryption [26] and full homomorphic encryption [27]. But these encryption techniques are not suitable for the current industrial needs. The proposed approach is an attempt to design the secure architecture for managing big data in cloud.

3. Secure Architecture for big data storage in cloud

In the cloud, data center is the major repository to store and manage the big data using the virtual or physical mode. Nowadays, many cloud service providers (CSPs) are managing the data centers to store the huge volumes of data. For instance, National weather forecasting center [28] is a public data centers which stores the huge volumes of weather data. Many organizations have their own private data centers to store their data frequently. Data centres plays vital role in performing complex computations and retrieving the large volumes of datasets from different sources.

Applications in the real time scenarios follow the distributed environment where it requires different types of datasets stored in different data centers of cloud. Therefore, it creates security and privacy issues at the time data storage and accessing. Big data is one of the recent advancements where the researchers are concentrating on developing the security approaches to preserve the data more confidential. This research work concentrates on developing the Big Data security architecture for cloud data centers which is shown in Figure 1. The proposed architecture consists of following modules:

3.1 Data collection module

In the data collection phase, the data is gathered from different sources such as user login transactional information, financial information, personnel information and also from other sources like customers, different data centers and external systems which are connected directly or indirectly to the proposed architecture.

3.2 Data processing module

Data processing is performed after the data collection module is completed. In the data processing module, user authentication and authorization will be completed by validating the user identity. Once the user identity is successfully verified, then the proposed model secures the login credentials. Public key integration (PKI) certificate methodology is used to secure the user credentials.

3.3 Data transfer Module

After completion of data processing phase, the user data is securely transferred to the different data centers present in different locations. In the proposed system, Meta cloud data storage architecture is used to transfer the data from applications to cloud and cloud to the applications.

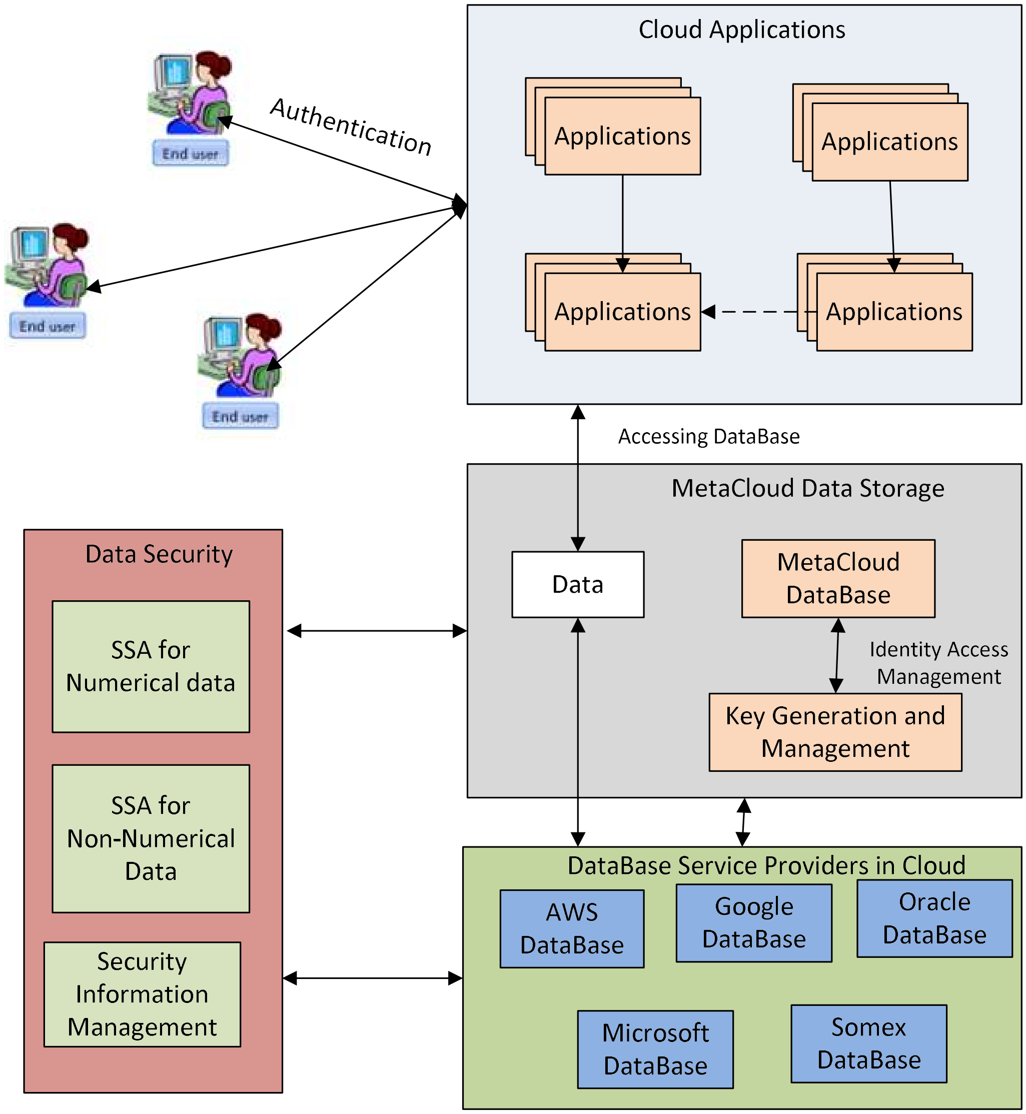

Figure 1: Big Data Security Framework for Cloud data centers

3.4 Data Categorization

In general, the data is classified into two categories, Normal and sensitive data. Normal data is a large volume of public data such as news and reports from the companies and organizations. This public data can be accessed by anyone and it is not required to secure the data. Sensitive data represents the wide range of data that contains the medical information, religious information, political information, military information, personal information and organization information etc. The sensitive data is further classified in to numerical data and non-numerical data. The numerical data contains the numerical values of the attributes whereas the non-numerical data contains the categorical or quantitative data that can be observed which is not measured.

3.5 Meta cloud data storage

Meta cloud data storage is used to secure the big data in the cloud environment. Various cloud data base service providers such as oracle, Google and Amazon cloud are used to store the user data. The normal and sensitive data is stored in to the data centers based on the priority levels. The key generation and management are used to develop the key and maintain a log for key management.

3.6 Data security

In the proposed framework, the data security plays crucial role in managing the user's data. The data security module contains the security service algorithms (SSA) for numerical data and non-numerical data. The algorithms are used to secure the data by performing the encryption or decryption techniques. The user has the privilege to select the algorithms based on the sensitivity of the data to store in the cloud.

4. Security Service Algorithm for Big Data

The proposed model is implemented using the architecture shown in figure 1. The input data is divided in to functional units while at the time of storage in the cloud. The proposed approach is developed to partition the sensitive data into numerical and non-numerical data by encrypting them for meta-storage in cloud. The Advanced RedistributiOn MetA storage (AROMA) algorithm is used to secure the data from cloud service providers without decreasing the performance.

4.1 Advanced RedistributiOn MetA storage (AROMA) algorithm

Figure 2 represents the high level workflow model of the proposed algorithm. The proposed algorithm has two stages. One is partitioning the input data into numerical and non-numerical data. The second one is to merge the partitioning data to obtain the original data.

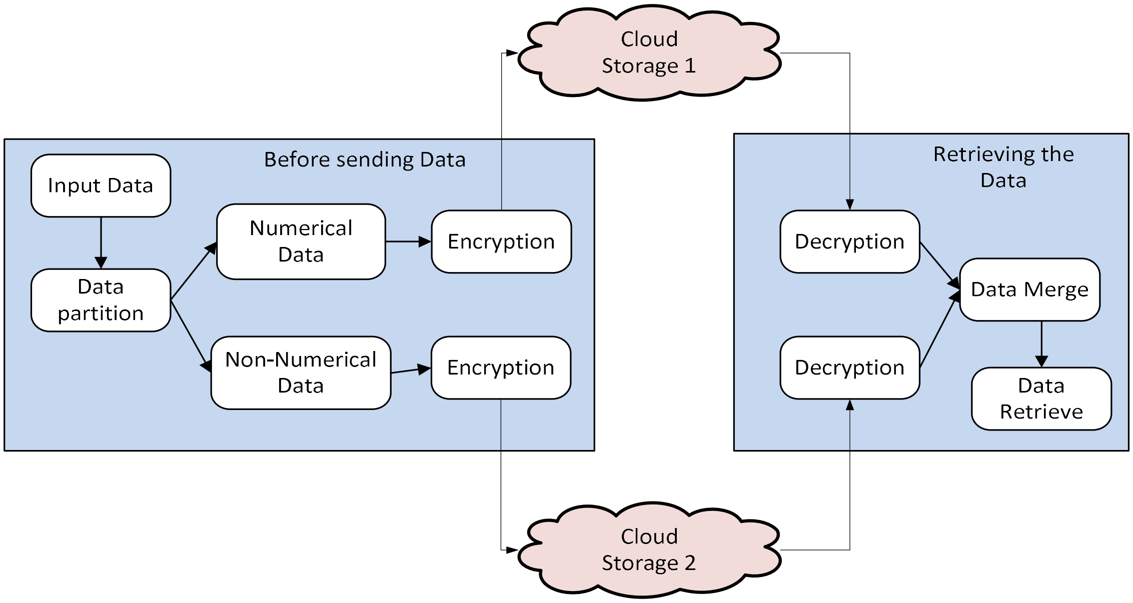

Figure 2: Workflow model for data partitioning and merging in the AROMA algorithm

AROMA algorithm is composed of encryption and decryption procedures for numerical and non-numerical data. Algorithm 1 shows the encryption procedure for securing the distributed data.

```
Algorithm 1: Encryption process
Input: U← Numerical/Non-Numerical Data
        R←Random binary parameter
```

```
Output: Encrypted data {A, B}
Begin
Step 1: Initialize $X \leftarrow 0$, $A \leftarrow 0$ and $B \leftarrow 0$
Step 2: Generate the random key K
Step 3: for all numerical/Non-Numerical data do
Step 4:    If $U \neq R$ && $R \neq 0$ then
Step 5:         Perform $X = U-R$
Step 6:         $A = R \oplus K$
Step 7:         $B = X \oplus K$
Step 8:    End if
Step 9: End for
End
```

In Algorithm 1, the input data is taken as either numerical data or non-numerical data which is represented as U and R is the random binary parameter to be a non-empty filed and it must be shorter and not equal to U. A and B are the encrypted data in cloud remote servers. An empty data set X is initialized along with A and B which are set to 0. A random key K is generated and stored in the users register which is used for encryption and decryption process. As a next step, calculate the value of X by subtracting the R from U and then perform two XOR operations to obtain the values of A and B that has to be stored in different cloud servers. An example is given in figure 3 to illustrate the procedure of Algorithm 1.

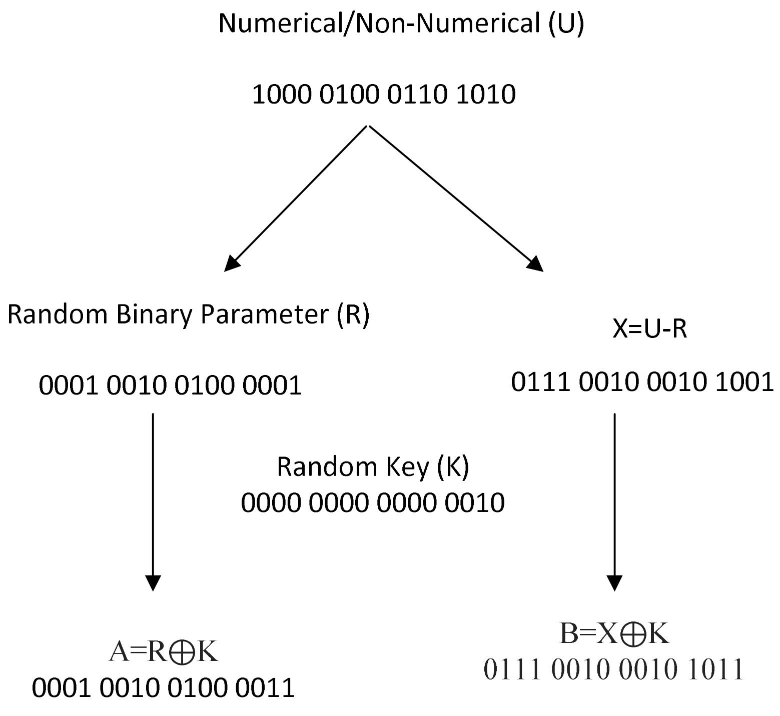

Figure 3: Encryption procedure for Numerical and Non-Numerical Data

In figure 3, the proposed model selects the user data U which is converted in to binary form. Then, the algorithm 1 generates the random binary parameter R as 0001 0010 0100 0001 and X can be calculated using the U-R as 0111 0000 0010 1001. As a next step, the random key K is generated as 0010 and performs the XOR operation with R and X. The values to be uploaded for A and B are 0001 0010 0100 0011 and 0111 0000 0010 1011 which is stored in the different cloud servers. By using this approach, the user data is more secure against the inside and outside attackers.

```
Algorithm 2: Decryption Process
Input: Encrypted data {A, B}, Random Key K
Output: U← Numerical/Non-Numerical Data
Begin
Step 1: Initialize $\alpha \leftarrow 0$, $\beta \leftarrow 0$ and $U \leftarrow 0$
Step 2: /* user receives encrypted data in A and B from Meta cloud */
Step 3: $\alpha = A \oplus K$
Step 4: $\beta = B \oplus K$
Step 5: $U = \alpha + \beta$
End
```

Algorithm 2 is used to retrieve the data from the cloud. The inputs of the algorithm are the encrypted data {A, B} along with the random key K. The user finds the key K from their registry. Algorithm 2 initializes the α, β data sets along with U which is set to 0 and it performs XOR operations to both A and B by applying the key K. the user data U is obtained by summation of α and β. Figure 4 shows an example for decrypting the user data.

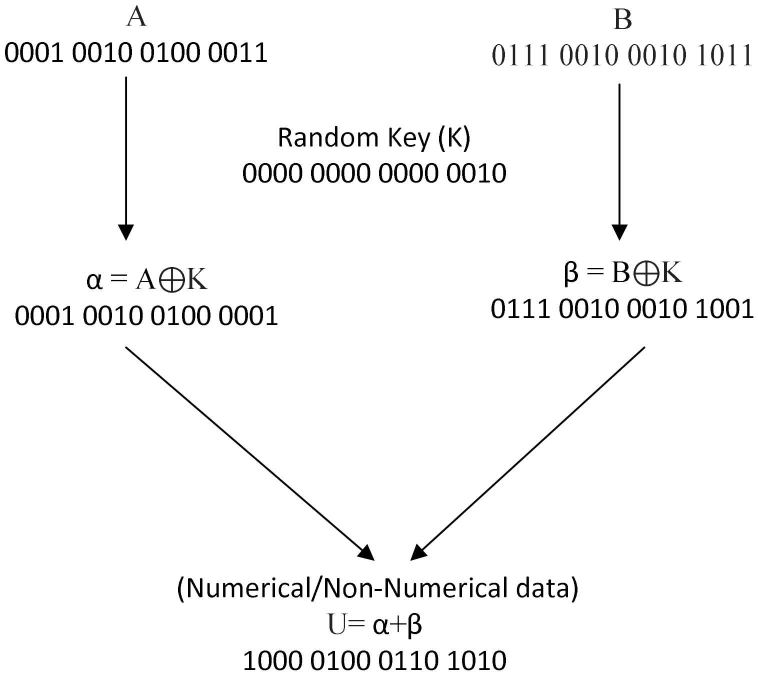

Figure 4: Decryption Procedure for Numerical/Non-Numerical Data

Figure 4 shows the retrieval of user data from the cloud servers. As a first step, user has to retrieve the encrypted data {A, B} from cloud servers. Then, the algorithm uses the XOR operation with the key K. the obtained binary values α and β are 0001 0010 0100 0001 and 0111 0010 0010 1001. Next, the sum operation is performed between α and β to obtain the user data 1000 0100 0110 1010.

5. Experimental setup and Result Analysis

This section represents the experimental setup and result analysis of the proposed model with respective of the execution time. The performance of the proposed method is compared with the existing algorithms like Data Encryption Standard (DES) [18], Blowfish algorithms [19] and Advanced Encryption Standard (AES) [17] algorithms.

To test the performance of the proposed model, we considered the real time environment called as Amazon EC2 [14] to store the user data. A virtual machine instance is created with 4-core CPU, 4-GB memory and 160 GB storage service. The Xen hypervisor is used to configure the virtual machine and windows server 2016 base is used as an operating system. A Hackman tool is installed in the virtual machine. This tool employs Brute force attack [15] and dictionary attack [16] on the data base.

In order to test the performance of the proposed model, some configurations had made to the data size and it is given in table 1.

Table 1: Data size configuration for AROMA

Data size	Encryption	Decryption
10MB	EConfig-1	DConfig-1
100MB	EConfig-2	DConfig-2
500MB	EConfig-3	DConfig-3

5.1 Result analysis

Figures 5, 6 and 7 show the comparison of the encryption execution time of DES, Blowfish, AES and AROMA algorithms. The experimental analysis was conducted on different configurations sizes such as EConfig-1, EConfig-2 and EConfig-3. In Figures 5, 6 and 7, it is observed that the proposed algorithm AROMA has shorter execution time compared to the DES, Blowfish and AES algorithms. In Figure 5, overall execution time of AROMA is 10% less compared to AES and in Figures 6 and 7 it is 18% and 21% less than AES execution time. It is due to the proposed mechanism follows the binary encryption with random generated keys and it seeks smaller time to convert the numerical or numerical data in to encryption format. In DES, it uses 64 bit blocks, which causes security issues at the time of encrypting the huge volumes of data with the same key and also it takes maximum execution time compared to other algorithms. Blowfish algorithm uses block cipher mode encryption. It uses more number of keys and it uses 64 bit block ciphers and the performance of the blowfish algorithm is better than the DES. AES is one of the standard symmetric encryption algorithms. AES holds the keys of sizes 128, 192 and 256 bits and it is proved as one of the secure algorithms for data encryption. But due to the key size length and complex procedure of encryption leads to the smaller depreciation compared to the AROMA algorithm.

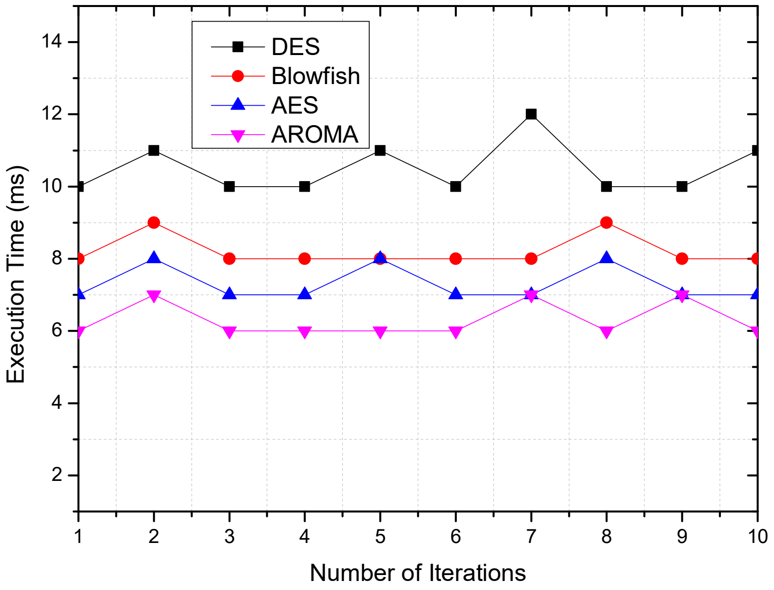

Figure 5: Comparison of the execution time under EConfig-1

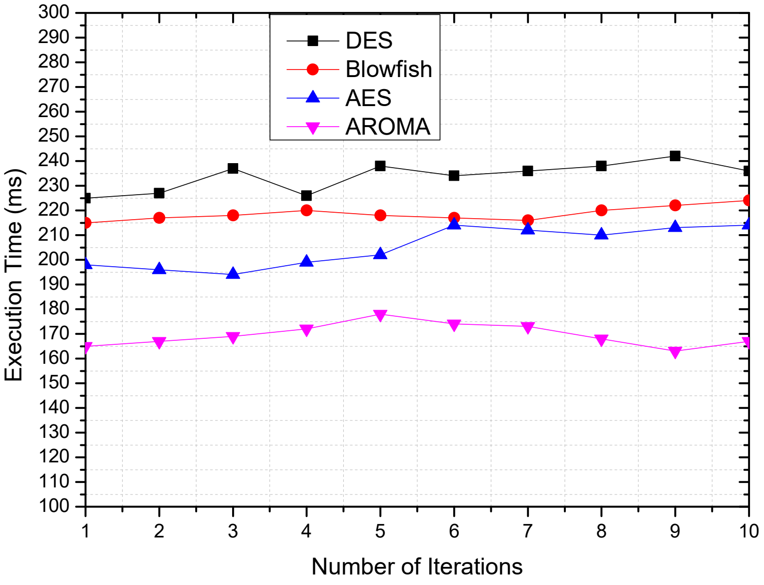

Figure 6: Comparison of the execution time under EConfig-2

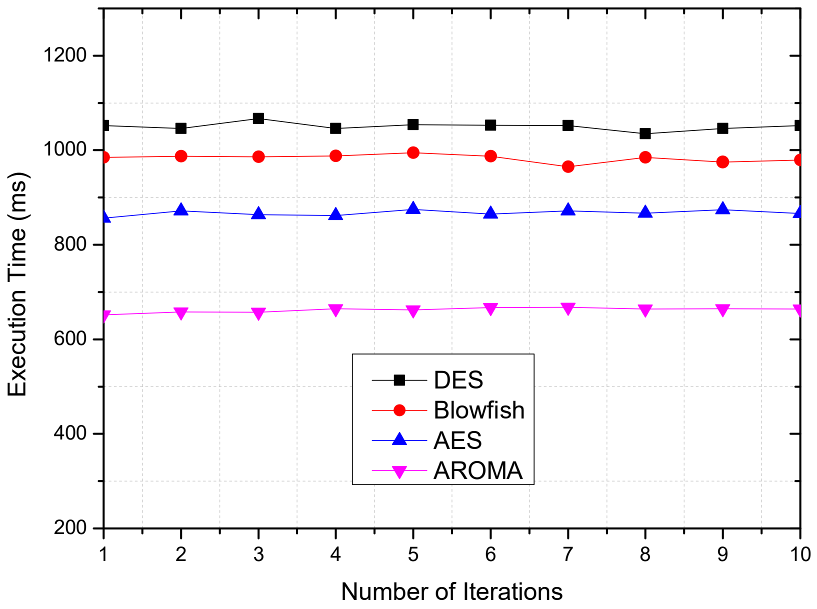

Figure 7: Comparison of the execution time under EConfig-3

Figures 8, 9 and 10 show the decryption execution time of the DES, Blowfish, AES and AROMA algorithms. The performance comparison of the proposed and existing algorithms is tested with different configurations such as DConfig-1, DConfig-2 and DConfig-3. In figures 8, 9 and 10, it is observed that the AROMA algorithm has minimum execution time compared to the DES, Blowfish and AES algorithms. In Figure 8, it is observed that AROMA has 8% less execution time compared to AES and it is recorded 11% and 17% less execution time compared to AES in Figures 9 and 10. Though AROMA have the better computing capacity, AES also served better in decryption than the DES and Blowfish algorithms.

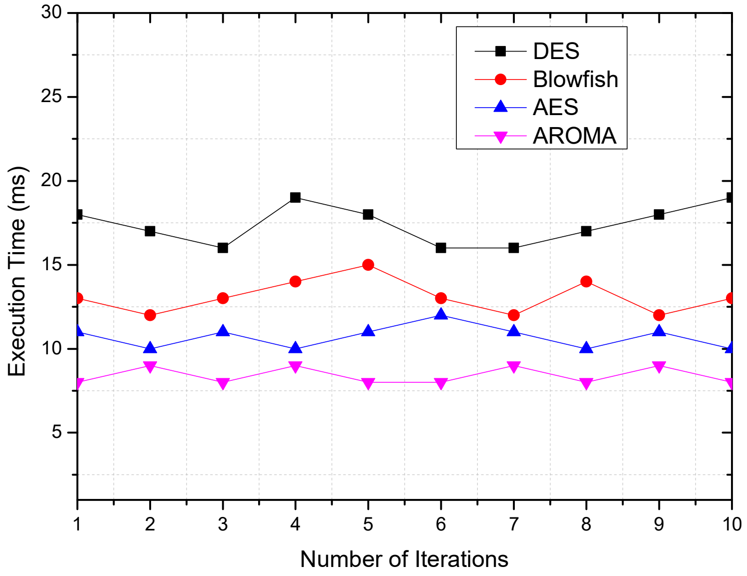

Figure 8: Comparison of the execution time under DConfig-1

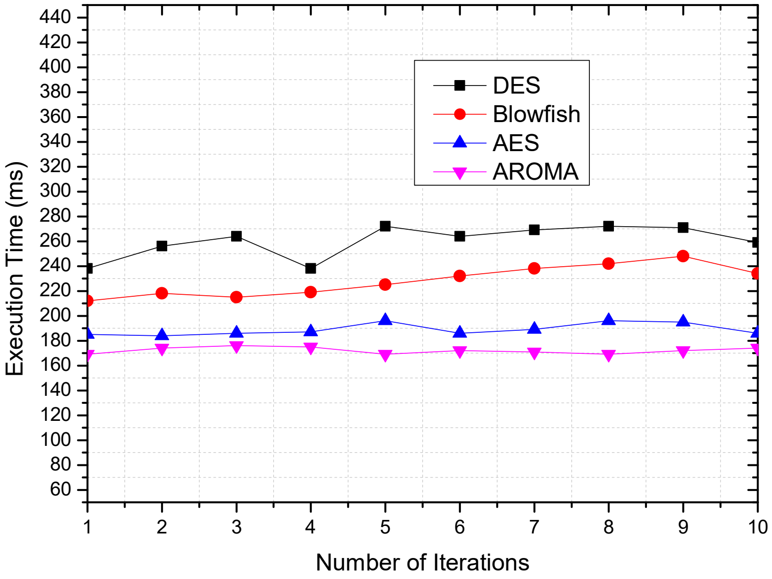

Figure 9: Comparison of the execution time under DConfig-2

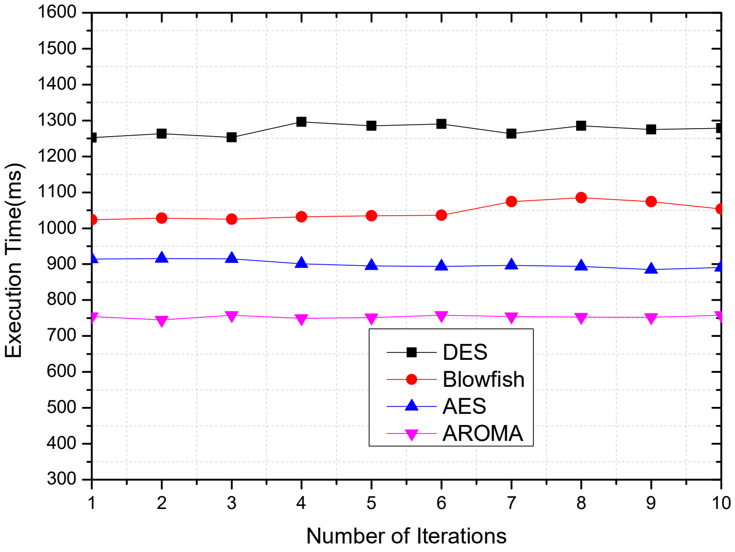

Figure 10: Comparison of the execution time under DConfig-3

6. Conclusion

This paper developed architecture for securing the big data in cloud. The proposed model considered the AROMA algorithm for encrypting and decrypting the user data. This would prevent the CSPs to gain control over the user data. The user data is divided in to numerical and non-numerical data. The encryption of the user data is performed before it is sending to the cloud and decryption is performed by accessing the encrypted data from Meta-cloud storage. The experimental results proved that AROMA algorithm is more secure and having less execution time compared to DES, Blowfish and AES algorithms. The Future work would address the security issues at the time of data duplications with respect to maximize the data availability.

References

[1] Cuzzocrea, Alfredo. "Privacy and security of big data: current challenges and future research perspectives." In *Proceedings of the First International Workshop on Privacy and Secuirty of Big Data*, pp. 45-47. ACM, 2014.

[2] Terzi, Duygu Sinanc, Ramazan Terzi, and Seref Sagiroglu. "A survey on security and privacy issues in big data." In *Internet Technology and Secured Transactions (ICITST), 2015 10th International Conference for*, pp. 202-207. IEEE, 2015.

[3] Puthal, Deepak, Surya Nepal, Rajiv Ranjan, and Jinjun Chen. "A synchronized shared key generation method for maintaining end-to-end security of big data streams." (2017).

[4] Storey, Veda C., and Il-Yeol Song. "Big data technologies and Management: What conceptual modeling can do." *Data & Knowledge Engineering* 108 (2017): 50-67.

[5] Hashem, Ibrahim Abaker Targio, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, and Samee Ullah Khan. "The rise of “big data” on cloud computing: Review and open research issues." *Information Systems* 47 (2015): 98-115.

[6] Assunção, Marcos D., Rodrigo N. Calheiros, Silvia Bianchi, Marco AS Netto, and Rajkumar Buyya. "Big Data computing and clouds: Trends and future directions." *Journal of Parallel and Distributed Computing* 79 (2015): 3-15.

[7] van Oosterom, Peter, Oscar Martinez-Rubi, Milena Ivanova, Mike Horhammer, Daniel Geringer, Siva Ravada, Theo Tijssen, Martin Kodde, and Romulo Gonçalves. "Massive point cloud data management: Design, implementation and execution of a point cloud benchmark." *Computers & Graphics*49 (2015): 92-125.

[8] Zhang, Yin, Meikang Qiu, Chun-Wei Tsai, Mohammad Mehedi Hassan, and Atif Alamri. "Health-CPS: Healthcare cyber-physical system assisted by cloud and big data." *IEEE Systems Journal* 11, no. 1 (2017): 88-95.

[9] C. Wang , S. Chow , Q. Wang , K. Ren , W. Lou , Privacy-preserving public auditing for secure cloud storage, IEEE Trans. Comput. 62 (2) (2013) 362–375.

[10] Wu, C., and Guo, Y. Enhanced User Data Privacy with Pay-By-Data Model. Proc. of BigData Conference, 2013.

[11] Li, M., Zang, W., Bai, K., Yu, M. and Liu, P. MyCloud: Supporting User-Configured Privacy Protection in Cloud Computing. Proc. of ACSAC, 2013.

[12] Kaur, M., & Singh, R. (2013). Implementing encryption algorithms to enhance data security of cloud in cloud computing. *International Journal of Computer Applications*, *70*(18).

[13] Arora, R., Parashar, A., & Transforming, C. C. I. (2013). Secure user data in cloud computing using encryption algorithms. *International journal of engineering research and applications*, *3*(4), 1922-1926.

[14] Juve, Gideon, Ewa Deelman, Karan Vahi, Gaurang Mehta, Bruce Berriman, Benjamin P. Berman, and Phil Maechling. "Scientific workflow applications on Amazon EC2." In *E-Science Workshops, 2009 5th IEEE International Conference on*, pp. 59-66. IEEE, 2009.

[15] Cho, Jung-Sik, Young-Sik Jeong, and Sang Oh Park. "Consideration on the brute-force attack cost and retrieval cost: A hash-based radio-frequency identification (RFID) tag mutual authentication protocol." *Computers & Mathematics with Applications* 69, no. 1 (2015): 58-65.

[16] Bellare, Mihir, David Pointcheval, and Phillip Rogaway. "Authenticated key exchange secure against dictionary attacks." In *international conference on the theory and applications of cryptographic techniques*, pp. 139-155. Springer, Berlin, Heidelberg, 2000.

[17] Lu, Chih-Chung, and Shau-Yin Tseng. "Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter." In *Application-Specific Systems, Architectures and Processors, 2002. Proceedings. The IEEE International Conference on*, pp. 277-285. IEEE, 2002.

[18] Mahajan, Prerna, and Abhishek Sachdeva. "A study of encryption algorithms AES, DES and RSA for security." *Global Journal of Computer Science and Technology* (2013).

[19] Devi, G. and Kumar, M.P., 2012. Cloud computing: A CRM service based on a separate encryption and decryption using Blowfish Algorithm. *International Journal Of Computer Trends And Technology*, *3*(4), pp.592-596.

[20] C. Wang, S. Chow, Q. Wang, K. Ren, W. Lou, Privacy-preserving public auditing for secure cloud storage, IEEE Trans. Comput. 62 (2) (2013) 362–375.

[21] Q. Liu, G. Wang, J. Wu, Time-based proxy re-encryption scheme for secure data sharing in a cloud environment, Inf. Sci. 258 (2014) 355–370.

[22] Y. Li, W. Dai, Z. Ming, M. Qiu, Privacy protection for preventing data over-collection in smart city, IEEE Trans. Comput. 65 (5) (2016) 1339–1350.

[23]] M. Ali, S. Khan, A. Vasilakos, Security in cloud computing: Opportunities and challenges, Inf. Sci. 305 (2015) 357–383

[24] C. Chen, C. Zhang, Data-intensive applications, challenges, techniques and technologies: A survey on big data, Inf. Sci. 275 (2014) 314–347

[25]] K. Gai, M. Qiu, L. Tao, Y. Zhu, Intrusion detection techniques for mobile cloud computing in heterogeneous 5G, Secur. Commun. Netw. (2015) 1–10.

[26] Waters, Brent. "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization." In *International Workshop on Public Key Cryptography*, pp. 53-70. Springer, Berlin, Heidelberg, 2011.

[27] Van Dijk, Marten, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. "Fully homomorphic encryption over the integers." In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 24-43. Springer, Berlin, Heidelberg, 2010.

[28] http://www.imd.gov.in accessed on 13/10/2017.

[29] Rajesh, N., and A. Arul Lawrence Selvakumar. "Hiding personalised anonymity of attributes using privacy preserving data mining." *International Journal of Advanced Intelligence Paradigms* 7, no. 3-4 (2015): 394-402.

[30] Sandhya, M. K., Krishnan Murugan, and P. Devaraj. "False data detection and dynamic selection of aggregator nodes with pair-wise key establishment in homogeneous wireless sensor networks." *International Journal of Advanced Intelligence Paradigms* 10, no. 1-2 (2018): 83-102.

[31] Raju, Dasari Naga, and Vankadara Saritha. "Architecture for fault tolerance in mobile cloud computing using disease resistance approach." *International Journal of Communication Networks and Information Security* 8, no. 2 (2016): 112.