International Journal on Emerging Technologies **11**(2): 584-589(2020)

ISSN No. (Print): 0975-8364
ISSN No. (Online): 2249-3255

A Novel Scheme for Security of Unstructured and Semi-Structured Big Data Outsourced to Cloud

P. Amarendra Reddy[1], P. Bhaskara Reddy[2] and O. Ramesh[3]

[1]Associate Professor, Department of Information Technology, MLR Institute of Technology (Telangana), India.
[2]Professor, Department of Electrical and Electronics Engineering,
Holy Mary Institute of Technology & Science (Telangana), India.
[3]Assistant Professor, Department of Computer Science and Engineering,
MLR Institute of Technology (Telangana), India.

(Corresponding author: P. Amarendra Reddy)

(Received 04 December 2019, Revised 30 January 2020, Accepted 05 February 2020)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: Big data is characterized with its attributes like Volume, Velocity and Variety. The characteristics of big data have in fact influence on the security mechanisms. From the literature, it is understood that one size does not fit for all. Both unstructured and semi-unstructured data can be secured with cryptographic primitives. However, structured data security can be leveraged with Homomorphic Encryption (HE). As big data is outsourced to cloud storage infrastructure, it is indispensable to safeguard it from internal and external attacks. Often sensitive data is stored in cloud storage facilities like Dropbox. Many security mechanisms based on cryptography were proposed earlier. There are certain security algorithms like RSA and AES widely used by cloud computing platforms. Due to the possible emergence of quantum computers, industry and academic are striving to improve security mechanisms. There are security challenges to existing encryption techniques when quantum computers emerged. Towards this end a novel security scheme known as Multi-Layered Hybrid Encryption Standard (ML-HES) is proposed. It is a hybrid approach that enhances big data security in cloud by meeting essential requirements of security known as confidentiality, data integrity and data availability. An empirical study is made with Amazon Elastic Compute Cloud (EC2) and S3 (Simple Storage Service) for enhanced security to outsourced big data. This research is confined to unstructured and semi-structured data. The results revealed that the proposed scheme outperforms the state of the art in terms of aforementioned security attributes.

Keywords: Immediately Big data, big data characteristics, security, cloud computing, hybrid encryption scheme.

I. INTRODUCTION

Big data is realized with the technology known as cloud computing. The rationale behind this is that there is much difference between traditional storage and cloud based storage. With cloud technology, Cloud Service Providers (CSPs) started investing on computing resources and the entire world can rely on it in pay as you go fashion instead of investing for Information Technology (IT) resources. This is actually paradigm shift between the traditional approach of IT and the cloud based one that we have realized in the recent past. This has revolutionized the way of storage and retrieval of data. Applications including mobile apps started performing well by offloading computing and storage processes to cloud. With scalable and available cloud infrastructure and distributed computing frameworks like Hadoop, there is an eco-system created to deal with big data storage. Big data has different characteristics such as Volume, Velocity and Variety. There are 3Vs out of which the first V indicates large volumes of data, the second V refers to the fact that the data is growing with some speed while the third V refers to the fact that the data is of different nature such as structured, semi-structured and unstructured.

Having understood big data characteristics, the study of literature revealed that security primitives being used are influenced by big data characteristics. For instance, same security algorithm or mechanism may not be good for all kinds of data or characteristics. We found that one size does not fit for all. For this reason, our research is divided into two parts. In the first part, semi-structured and unstructured data (variety feature) along with voluminous data are considered for empirical study. Our research with structured data and integration of the first and second are deferred to our future work. Therefore, this paper focuses on big data security by considering semi-structured and unstructured data. The former is having features of structured data but stored in a flat file while the latter lacks features of structured data and the data is stored in flat files. Any textual document is an example for unstructured data while the data in the form of XML is often treated as semi-structured data as the data has some structure or it may be as good as a database.

As the cloud – big data eco-system emerged, there are many security threats identified. When data owners outsourced their data in plain text, it may be subjected to internal and external attacks. Once the data is outsourced to CSP premise, the data owner has no control over it. It is the main cause of concern. There are many reasons for data loss such as internal or external attacks, intentional removal of data by CSP, hidden hardware faults and accidental removal of data due to manual mistakes. Therefore, there are many security challenges to be addressed. The existing schemes like AES and RSA which are widely used by cloud service providers fail if the hackers exploit the possible emergence of quantum computing. Many existing cloud based solutions are still using the traditional algorithms. This causes problems in future. Therefore, there are many security challenges to be addressed. The existing schemes like AES and RSA which are widely used by cloud service providers fail if the hackers exploit the possible emergence of quantum computing. Therefore, industry and academia are continuously trying to improve the state of the art. Even quantum cryptography is being worked out. In the context aforementioned, we proposed a novel security scheme for big data security. It will enable higher level of security as it has multiple layers. Besides it makes the adversaries hard to break it. When compared with RSA and AES, the proposed method showed better performance. Our contributions are as follows.

– A novel security scheme is designed with multiple objectives such as confidentiality, integrity and availability in mind.

– A prototype is built for secure outsourcing of big data to public cloud (Amazon EC2) and retrieval for showing effectiveness of the proposed security scheme.

– The scheme is evaluated with the state of the art and found to be highly secure.

The remainder of the paper is structured as follows. Section II reviews literature on the security schemes used for big data security. Section III presents the proposed scheme and its functionalities. Section IV provides experimental setup. Section V presents results of security analysis of the proposed scheme and its comparison with existing ones. Section VI concludes the paper and provides scope of future work.

II. RELATED WORKS

Many security schemes came into existence for cloud data security. Especially for securing big data AES and RSA are widely used. Lee *et al.,* (2018) used AES for cloud data security under the cloud named Heroku. Their security mechanism divides a file into number of blocks and encrypts it prior to sending to cloud. They found that AES was capable of securing data [1]. Tamilselvi (2017) also used AES for data security in cloud. They found the utility of the AES for cloud based storage [2]. Delfin *et al.,* (2018) proposed a system for outsourcing data with security using AES based solution. They described system with multiple modules implemented. The application of AES with the cloud storage was found to be effective and reliable [3]. Emdad and Khan (2019) also investigated AES and its modus operandi in order to have better security to cloud storage. They proposed a methodology for systematic application of security to the big data being outsourced to cloud [4]. Babitha and Babu (2016) used 128 bit AES for cloud storage. They found that AES was faster than other security schemes [5].

The asymmetric cryptography known as RSA is also used in different cloud applications. Parthasarathy *et al.*, (2019) used it for secure data storage and retrieval in cloud computing. Their methodology includes the procedures involved in RSA for higher level of security. Since it is a public key infrastructure, it was able to eliminate the need for key sharing [9]. Singh *et al.,* (2016) also employed RSA cryptography for cloud. They investigated time and space complexity with RSA. They found that time complexity is linearly increased based on key length [10]. Soumya and Prabha (2015) used RSA for data security in cloud [11]. Thilagavathy and Murugan (2017) on the other hand enhanced RSA for cloud storage security [12]. They improved RSA with longest common sub sequence of a string. They found it to be more secure than traditional RSA. Dalwal (2019) made a review of many security algorithms being used for cloud computing to protect communications with different cloud services [13].

Kumar *et al.*, (2016) proposed a hybrid algorithm based on RSA and AES. They opined that the hybrid approach cloud improves security level when employed to data stored in cloud [6]. Bhute and Arjaria (2016) made hybrid algorithm based on AES and RC6. With this they opined that they could realize a secure cloud framework. They also used the concepts of bit rendering and notification for handing security threats. With combined approach they found higher level of security [7]. Pius *et al.,* (2018) combined two algorithms known as Blowfish and AES. They employed these two methods in combination to secure storage and retrieval in cloud. They could prevent data leakage and data loss with their approach [8]. Selvamani and Rao (2017) used RSA and AES together for secure data sharing. They implemented an application with the hybrid approach to show the utility of the security enhancement [14]. Abdulkarim and Souley (2017) proposed a security system for cloud storage using RSA and digital signature. They found it more secure with the combined approach [15]. Elliptic curve based security is provided in [16]. Cyber security threats and prevention measures are found in [17]. From the literature, it is understood that the RSA and AES based methodologies suffer from limitations individually. However, when combined with other algorithms, their security gets enhanced. Nevertheless, in the possible emergence of quantum computers, researchers and academia are striving towards higher level of security. This is the research gap and this paper has taken a step towards more secure solution.

III. PROPOSED SCHEME

When it comes to big data and its security in cloud, there are many challenges associated with the eco-system. Out of the challenges, security is one of the big concerns to data owners. Many cryptography techniques are used to encrypt data before outsourcing to public cloud. However, most of the techniques are on the verge of being obsolete with the possible emergence of quantum computing which will be exploited by adversaries. In order to address this problem, a novel security scheme known as Multi-Layered Hybrid Encryption Standard (ML-HES) is proposed and implemented. This scheme is meant for secure storage and retrieval of big data. It ensures data availability, data integrity and confidentiality besides preventing data leakage. As the existing AES and RSA may be vulnerable in the context of quantum computing, a multi-layered and hybrid approach is followed. When AES (symmetric block cipher) is used alone, it may not be able to withstand aforementioned scenarios. Therefore, we used it as one of the layers in the proposed scheme. AES with 256 bits' key is used as the first layer of defense. Then Information Dispersal Algorithm (IDA) along with error correction code known as Reed-Solomon is used for achieving confidentiality. Then SHA-512 is used along with IDA slices to achieve data integrity. The data slices made with IDA help in achieving data availability.

Table 1: Notations used in the proposed scheme.

Change	Description
D	Denotes original data
$D^{'}$	Denotes encrypted data
$D^{''}$	Set of slices
$D^{'''}$	Final encoded data
d_i	Data file with slices
δ, Δ	Denote encryption and decryption respectively
sk	Secret key
$d_i^{'} = H(d_i)$	Application of hash value
ǁ	Operator for concatenation
E=$\|D^{'}\|$	Length of encrypted file
p	Denotes number of slices
q	number of symbols/bytes required to reconstruct $D^{'}$
ℓ= $\|d_i\|$	Slice data file length
$Y_i = Y_{(i-1)m+1} \ldots.. \; Y_{i.m}$	Denotes ith blocks of m symbols
θ	Original matrix
d_i= $d_{i1}, d_{i2}, \ldots\, d_{i\ell}$	File slice
C	Cauchy matrix

As presented in Table 1, the notations are provided along with the description to understand the proposed scheme.

Encoding Procedure: The proposed scheme has both encoding and decoding procedures. Both are part of the novel security scheme known as Multi-Layered Hybrid Encryption Standard (ML-HES)

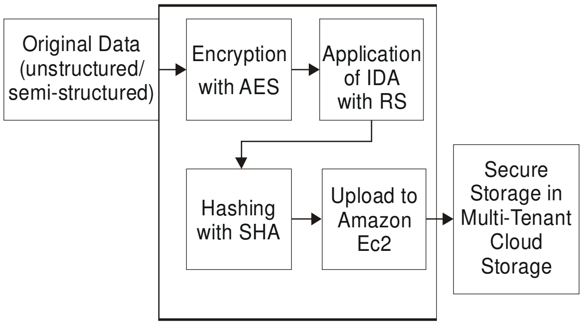
Original Data (unstructured/ semi-structured)
Encryption with AES
Application of IDA with RS
Hashing with SHA
Upload to Amazon Ec2
Secure Storage in Multi-Tenant Cloud Storage

Fig. 1. Secure data outsourcing with the encoding process.

As presented in Fig. 1, there are many mechanisms involved in the encoding process prior to outsourcing data to cloud. Multi-level security is achieved with the encoding process. First of all, the given data is subjected to generation of block cipher using AES with 256-bit length key. The key is generated dynamically and managed. Then IDS is employed to generate n number of slices from encrypted data. The slices are generated in such a way that only m numbers of slices out of n are sufficient to reconstruct data. This feature is crucial for promoting data availability in cloud. For each slice hash value is computed using the SHA-512 standard. The result of this operation is then uploaded to cloud for secure storage. When a cloud server goes down, it can reconstruct from the slices available in other servers.

As presented in the Listing 1, the encoding procedure takes given data, secret key and threshold in terms of m and n as input. Then it produces the finally encoded data that is ready to be outsourced to public cloud.

Listing 1: Pseudo code for encoding procedure.

```
Pseudocode for Encoding Procedure
Inputs: Data D, secret key sk, threshold value (m,n)
Output: Encoded data D'''
    Initialize D'
    Initialize D''
    Initialize D'''
    D'=ApplyAES(D, sk)
    D''=ApplyIDS(D', m, n)
    For each $d_i$ in D''
        $d_i$'=ApplySHA($d_i$)
        $d_i$''=Concatenate($d_i$, $d_i$')
        Add di'' to D'''
    End For
    Return D'''
```

$$\theta_{q\times\ell} = \begin{pmatrix} & Y_1\, Y_{q+1} \ldots.. & Y_{(\ell-1).q+1} \\ & Y_1\, Y_{q+1} \ldots.. & Y_{(\ell-1).q+1} \\ . & . & . \\ . & . & . \\ . & . & . \\ & Y_q\, Y_{2q} \ldots.. & Y_{\ell.q} \end{pmatrix}; \ell = \left(\frac{E}{q}\right). \qquad (1)$$

The file which is used as input is denoted as $D^{'}$ and the Eqn. 1 shows *q*x*l* matrix is provided where the size of data slice is denoted as *l*. The matrix is then transformed as shown in Eqn. 2.

$$C_{p\times q} = \begin{pmatrix} a_{11} & a_{12} \ldots. & a_{1q} \\ a_{21} & a_{22} \ldots. & a_{2q} \\ . & .\;\; . & . \\ . & .\;\;\; . & . \\ . & .\;\;\;\; . & . \\ a_{p1} & a_{p2} \ldots. & a_{pq} \end{pmatrix}. \qquad (2)$$

Each subset of rows (q) is a vector and the multiplication as shown in Eqn. 3 is needed to achieve dispersal.

C. θ =

$$\begin{pmatrix} a_{11} & a_{12} \ldots. & a_{1q} \\ a_{21} & a_{22} \ldots. & a_{2q} \\ . & .\;\; . & . \\ . & .\;\;\; . & . \\ . & .\;\;\;\; . & . \\ a_{p1} & a_{p2} \ldots. & a_{pq} \end{pmatrix} . \begin{pmatrix} & Y_1\, Y_{q+1} \ldots.. & Y_{(\ell-1).q+1} \\ & Y_1\, Y_{q+1} \ldots.. & Y_{(\ell-1).q+1} \\ . & . & . \\ . & . & . \\ . & . & . \\ & Y_q\, Y_{2q} \ldots.. & Y_{\ell.q} \end{pmatrix}$$

$$\begin{pmatrix} f_{11} & f_{12} \cdots & f_{1\ell} \\ f_{21} & f_{22} \cdots & f_{2\ell} \\ \cdot & \cdot \quad \cdot & \cdot \\ \cdot & \cdot \quad \cdot & \cdot \\ \cdot & \cdot \quad \cdot & \cdot \\ f_{p1} & f_{p2} \cdots & f_{p\ell} \end{pmatrix} = \delta_{p\times\ell}$$

When the dispersal algorithm is employed, the D' is transformed into D''as shown in Eqn. 3.

$$D'' = (d_{11,} \ldots, d_{1\ell}), \ldots , (d_{i1,} \ldots, d_{i\ell}), \ldots , (d_{p1,} \ldots, d_{p\ell}). \quad (3)$$

$$D''' = d_1 \parallel d_1', d_2 \parallel d_2', \ldots, d_p \parallel d_p'. \quad (4)$$

Once D'' is obtained, it is subjected to hashing in order to produce final slices denoted as D'''that can be uploaded to cloud.

Decoding Procedure: The decoding process is employed when data needs to be verified for integrity and downloaded from the cloud storage. It is actually the reverse procedure to encoding procedure described in Section 3.1.

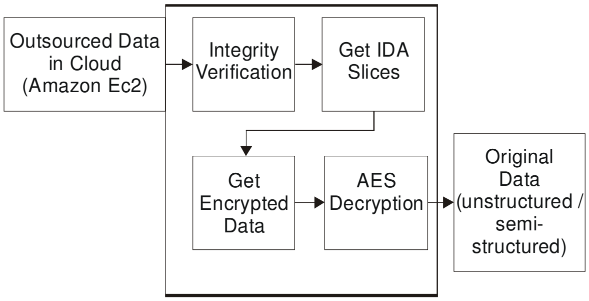
Outsourced Data in Cloud (Amazon Ec2)
Integrity Verification
Get IDA Slices
Get Encrypted Data
AES Decryption
Original Data (unstructured / semi-structured)

Fig. 2. Secure data retrieval with the decoding process.

As shown in Fig. 2, in the decoding process data verification is involved to check integrity. First of all, integrity of the outsourced data is verified prior to performing other operations. Once integrity is proved, the rest of the decoding process continues. If not, the missing slices or corrupted slices get reconstructed. After verification is completed, the process of IDA is employed in order to get the data and finally get encrypted data. After this process, the AES is employed in order to decrypt data and obtain original data.

Listing 2: Pseudo code for decoding procedure.

```
Pseudo code for Decoding Procedure
Inputs: Outsourced data D''', secret key sk, value for m
Output: Data D
    Initialize D'
    Initialize D''
    Initialize D'''
    For each d_i in D'''
    IF data integrity check is successful Then
    d_i=Deconcatenate(d_i'')
        Append data to D''
    Else
    Recover data
      End If
    End For
    D'=ApplyIDA(D'')
    D=ApplyAES(D')
    Return D
```

IV. EXPERIMENTAL SETUP

Experiments are made with a prototype application developed using Java platform. The application provides web based interface to interact with Amazon EC2 cloud where S3 is used for storing unstructured and semi-structured data.

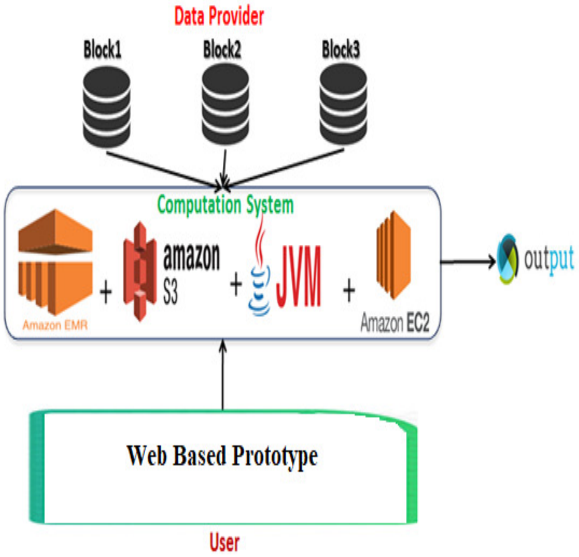

Fig. 3. Overview of the experimental environment.

As presented in Figure 3, the experimental setup is made with the prototype application built and the components of Amazon cloud. Amazon S3 is used for storage service. Amazon EC2 is the cloud instance where cluster is created. Web based application is used to execute the proposed scheme with features related to data encoding and decoding.

V. EXPERIMENTAL RESULTS

Experiments are made with the proposed scheme implemented as part of a web based application. As illustrated in Section IV, the Amazon cloud is used for empirical study. Amazon S3 is the actual storage service for both unstructured and semi-structured data. The results are observed in terms of data integrity, availability and confidentiality. However, the execution time is the metric used for presenting the performance of the proposed scheme named ML-HES and compared it with traditional AES and RSA.

Table 2: Execution time performance comparison for encoding/encryption process.

Data Size (MB)	Execution Time for Encoding/Encryption Process (seconds)		
	RSA	AES	ML-HES
10	2.9838	0.8989	1.9057
50	4.6542	2.4956	3.5537
100	5.4915	2.6968	3.8537
500	25.3956	12.9996	14.7537

As presented in Table 2, the execution time taken for encoding or encryption by algorithms like RSA, AES and the proposed ML-HES is provided against different data sizes.

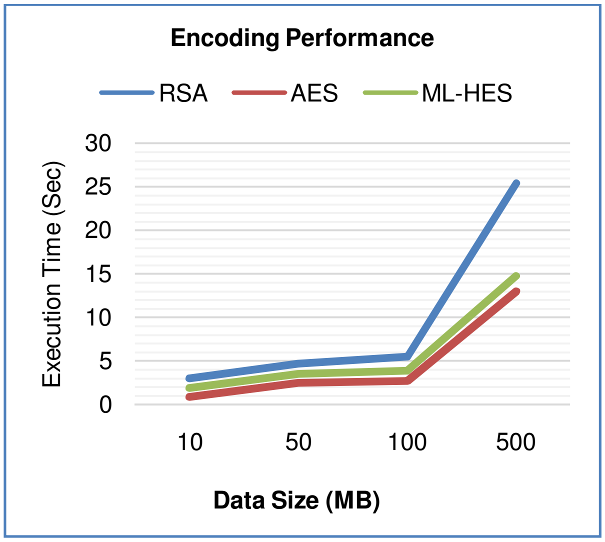

Fig. 4. Performance of the proposed scheme for encoding.

As presented in Fig. 4, it is understood that the data of different size is used for experiments. It includes 10 MB, 50 MB, 100 MB and 500 MB. As there is increase in the data the execution time is increased linearly. Another important observation is that the traditional AES performed better then ML-HES. However, AES and ML-HES are proved to be better than RSA. The performance of ML-HES is less in terms of execution time, as it provides multi-level security which is better than that of AES. When data size is 10 MB, the decoding time taken by RSA is 2.9838 seconds, AES 0.8989 seconds and ML-HES 1.9057 seconds. When the data size is 500 MB, RSA took 25.3956 seconds, AES 12.9996 seconds and ML-HES 14.7537 seconds.

Table 3: Execution time performance comparison for decryption/decoding.

Data Size (MB)	Execution Time for Decoding / Decryption (seconds)		
	RSA	AES	ML-HES
10	2.7891	0.6851	1.5985
50	3.9396	1.8995	2.5989
100	4.8294	2.0196	2.7862
500	18.983	9.1252	9.9724

As presented in Table 3, the execution time taken for decoding or encryption by algorithms like RSA, AES and the proposed ML-HES is provided against different data sizes.

As presented in Fig. 5, it is understood that the data of different size is used for experiments. It includes 10 MB, 50 MB, 100 MB and 500 MB. As there is increase in the data the execution time is increased linearly. Another important observation is that the traditional AES performed better then ML-HES. However, AES and ML-HES are proved to be better than RSA. The performance of ML-HES is less in terms of execution time, as it provides multi-level security which is better than that of AES. When data size is 10 MB, the decoding time taken by RSA is 2.7891 seconds, AES 0.6851 seconds and ML-HES 1.5985 seconds. When the data size is 500 MB, RSA took 18.983 seconds, AES 9.1252 seconds and ML-HES 9.9724 seconds.

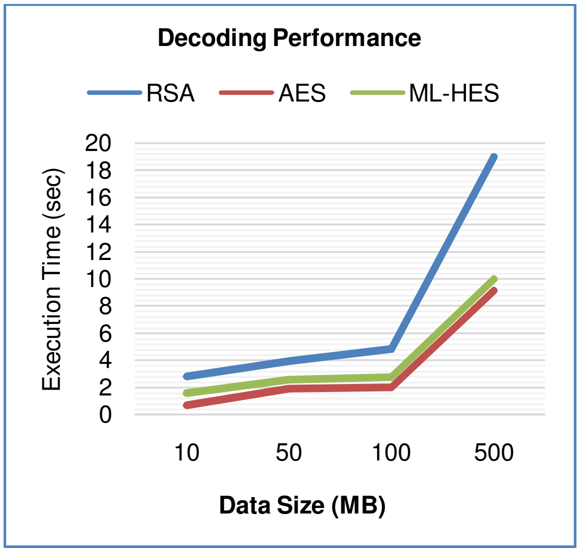

Fig. 5. Performance of the proposed scheme for decoding.

VI. CONCLUSION AND FUTURE WORK

In cloud storage infrastructure the big data outsourced may be stored in different servers. As the resources are shared in multi-tenant environment, the cloud infrastructure is vulnerable to insider and outsider threats. As the cloud storage infrastructure is not under direct control of the consumers or organizations, it is treated as untrusted. Data owners often hesitate in outsourcing data due to security reasons. The existing cryptographic solutions such as RSA and AES become vulnerable with the possible emergence of quantum computers. Therefore, there is need for enhanced security, confidentiality, data integrity and data availability. Towards this end, a novel security scheme is proposed in this paper. It has encryption and decryption mechanism along with hashing and information dispersal mechanism. It is known as Multi-Layered Hybrid Encryption Standard (ML-HES) which provides more depth in security in terms of encoding and decoding. Moreover, it also provides security to unstructured and semi-structured big data in terms of confidentiality, integrity and availability. With hashing mechanism, there is verification of data integrity in the decoding process. The encrypted data is outsourced to multiple cloud service providers (in a federated cloud) and the recovery of original data is guaranteed. Amazon EC2 used as cloud platform and S3 is used for big data storage and retrieval with enhanced security. In future, we deal with structured data (characteristic of big data) with flexible encryption mechanism besides making a hybrid approach that serves well in dealing with all characteristics of big data.

ACKNOWLEDGEMENTS

The authors would like to acknowledge Science and Engineering Research Board, India for financial support.

Conflict of Interest. No.

REFERENCES

[1]. Lee, B. H., Dewi, E. K. and Wajdi, M. F. (2018). Data security in cloud computing using AES under

HEROKU cloud. *Wireless and Optical Communication Conference (WOCC),* 1-5.

[2]. Tamilselvi, S. (2017). Data Storage Security in Cloud Computing Using AES. *International Journal of Advanced Networking & Applications*, *8*(5): 124-127.

[3]. Delfin, S., Sai, R. B., Meghana, J. V., Kundana L. Y., & Sharma, S. (2018). Cloud Data Security Using AES Algorithm. *International Research Journal of Engineering and Technology, 5*(10), 1189-1192.

[4]. Embad, M. R. B., & Khan, M. S. (2019). A Standard Data Security Model Using AES Algorithm in Cloud Computing. *International Journal of Software & Hardware Research in Engineering, 7*(5), 49-53.

[5]. Babitha M. P., & Babu, K. R. R. (2016). Secure cloud storage using AES encryption. *International Conference on Automatic Control and Dynamic Optimization Techniques,* 859-864.

[6]. Kumar, B., Boaddh, J., & Mahawar, L. (2016). A hybrid security approach based on AES and RSA for cloud data. *International Journal of Advanced Technology and Engineering Exploration, 3*(17), 43-49.

[7]. Bhute, S., & Arjaria, S. K. (2016). An efficient AES and RC6 based cloud-user data security with attack detection mechanism. *International Journal of Advanced Technology and Engineering Exploration, 3*(21), 110-117.

[8]. Pius, U. T., Onyebuchi, E. C., Chinasa, O. P. and Adoba, E. F. (2018). A Cloud-Based Data Security System using Advanced Encryption (AES) and Blowfish algorithms. *Journal of Scientific and Engineering Research*, *5*(6), 59-66.

[9]. Parthasarathy, R., Yee, H. W., Loong, S. S., Rajamanickam, L., & Ayyappan, P. (2019). *Implementation of RSA Algorithm to Secure Data in Cloud Computing, 6*(4), 61-68.

[10]. Singh, S. K., Manjhi, P. K., & Tiwari, R. K. (2016). Data Security using RSA Algorithm in Cloud Computing. *International Journal of Advanced Research in Computer and Communication Engineering*, *5*(8), 11-16.

[11]. Soumya, N. S. and Prabha, R. (2015). Cloud Computing: Data Security Using RSA. *IJLTEMAS*, *4*(10), 57-59.

[12]. Thilagavathy, R., & Murugan, A. (2017). Secure the Cloud Data Transmission using an Improved RSA Algorithm. *Indian Journal of Science and Technology, 10*(12), 1–6.

[13]. Chandrika, & Dalwal, E. S. (2019). Data Security in Cloud Computing Using Cryptographic Algorithms: A Review. *International Journal of Innovative Research in Computer and Communication Engineering*, *7*(1), 89-94.

[14]. Selvamani, K., & Rao, V. R. (2017). RSA and AES Based Secure Data Sharing in Cloud Based Environment. *International Journal of Computing, Communications & Instrumentation Engg.*, *4*(1), 111-114.

[15]. Abdulkarim, A. I., & Souley, B. (2017). An Enhanced Cloud Based Security System Using RSA as Digital Signature and Image Steganography. *International Journal of Scientific & Engineering Research, 8*(7), 1-6.

[16]. Bhardwaj, K., & Chaudhary, S. (2012). Implementation of Elliptic Curve Cryptography in ‘C’. *International Journal on Emerging Technologies, 3*(2), 38-51.

[17]. Bhardwaj, M., Fageria, A., Sain, N., & Kumar, B. (2018). Security Threats & Their Solution to Prevent Against Cyber Attacks. *International Journal of Electrical Electronics & Computer Science Engineering*, 122-125.

How to cite this article: Reddy, P. A., Reddy, P. B. and Ramesh, O. (2020). A Novel Scheme for Security of Unstructured and Semi-Structured Big Data Outsourced to Cloud. *International Journal on Emerging Technologies*, *11*(2): 584–589.