Flexible And Efficient Encryption Scheme For Data Dynamics On Encrypted Outsourced Data To Public Cloud

D. Ramesh,B. Rama

Abstract: -Persistence of data has been around for many decades. In computer science, storing data permanently and efficiently is the rationale behind all the innovations in the persistence media. The contemporary technology to store and handle voluminous data is the storage service rendered by cloud computing platforms. However, data owner has no control over cloud infrastructure. It is totally in an untrusted environment. Nevertheless, there have been some mechanisms to deal with the security of data that has been outsourced. Right from public auditing methods to provable data possession to data integrity methods came into existence to ensure cloud storage security. Cryptographic methods could solve security problems. However, flexible encryption methods that help in secure data outsourcing that supports search and data dynamics on encrypted data are still desired. In this paper, we proposed a methodology that handle secure data storage in cloud and allow operations directly on encrypted data. Besides it takes care of the different formats of data. Jelastic cloud platform is used with two kinds of environments, one for structured data and one for unstructured and semi-structured data, are used for empirical study. A prototype application is built using Java programming language that facilitates intuitive interface to have flexible encrypted storage in public cloud and data dynamics directly on encrypted cloud datawith the help of Homomorphic Encryption (HE). The empirical results showed that the proposed system is more efficient than many existing methods.

Keywords : Cloud computing, storage security, flexible encryption, homomorphic encryption, Jelastic cloud

1. INTRODUCTION

Security mechanisms are essential when data is outsourced to cloud. Lack of security results in deterioration of performance and the user base as well. Since data of data owners is stored in the public cloud provided by an organization known as Cloud Service Provider (CSP) [34], there are many apprehensions on the security of data being outsourced to cloud. Such security concerns are understandable as the cloud is untrusted environment and it is not under control of data owners. The issues involved with storage security are identified and presented in Figure1.

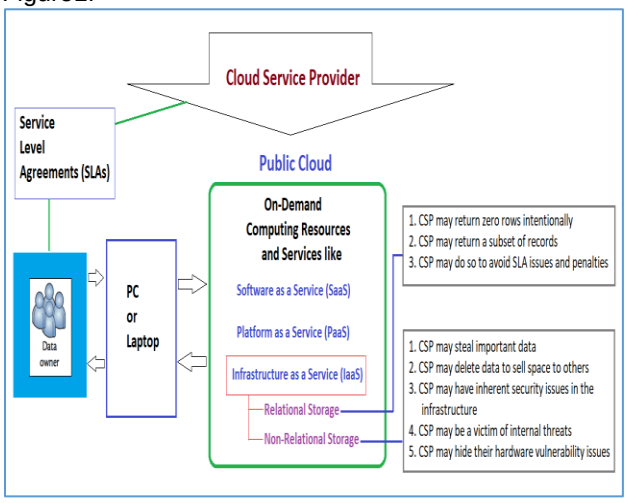

Figure 1:*Shows the problem context related to outsourcing data to public cloud*

- *D. Ramesh is currently pursuing Phd program in department of Computer ScienceKakatiya University Warangal Telangana State, India, E-mail: rameshd525@gmail.com*
- *B. Rama, assistant professor in department of Computer ScienceKakatiya University Warangal Telangana State, India. E-mail: rama.abbidi@gmail.com*

From the literature on the cloud computing and the security issues [15], it is understood that there are many reasons to have a security framework that takes care of structured, semi-structured and unstructured data. As presented in Figure 1, both non-relational and relational data needs to be outsourced to cloud. Moreover,three specific problems on relational storage and five specific issues with non-relational data are provided. All the kinds of data need secure storage and retrieval besides operations on encrypted data directly. Many solutions to this problem exist in literature. The concept of homomorphicencryption and fully HE are found to have attracted considerable interest among academia and researchers. HE is the main focus in [2], [4], [6], [14] [16] and [33]. Fully homomorphic cryptography (FHE) is explored in [10]. From the literature, it is found that there is no comprehensive framework that takes care of both structured and unstructured data with an efficient, flexible and secure storage framework that supports search and data dynamics on the encrypted data. Thus HE and FHE based solutions provided valuable insights to effectively utilize cloudplatforms. Nevertheless, the drawbacks of existing solutions overcome in this paper. Its contributions are as follows.

1. It presents a framework for efficient, flexible storage and retrieval of data in public cloud. In fact, it allows search operations and data dynamics on the outsourced data which is in encrypted. It also facilitates data modifications directly on the encrypted data.
2. An algorithm is proposed based on FHE for supporting the intended operations on encrypted data.
3. A prototype application s developed to show the performance of the proposed system. The results revealed that the proposed system is more flexible, secure and supports data dynamics and search operations on the outsourced encrypted data.

The remainder of the paper is structured as follows. Section 2 provides review of literature. Section 3 provides general

procedure for secure cloud storage. Section 4 presents the proposed secure storage and retrieval framework. Section 5 presents experimental setup. Section 6 presents results of experiments. Section 7 concludes the paper and gives directions for future work.

2. RELATED WORK

This section provides review of literature on cloud data encryption and HE. It provides insights on the kind of encryption that allows search operations on the encrypted data. Zhao et al. [1] proposed FHE based solution for data security. The principle of FHE is discussed and used. Tebaa et al. [2] exploited HE to secure cloud data.Tebaa and Hajii [3] and Tebaa et al. [4] also studied HE for cloud storage. It was multiplicative approach for encryption and decryption. Yu et al. [5] used cryptographic methods to obtain top k results. It dealt with statistic leakage, similarity relevance, robustness of the scheme, and the design considerations. It could guarantee data privacy. Potey et al. [6] focused on HE for security of content of MongoDB. Kokabas and Soyota [7] employed HE for Medical Cloud Computing. They used Amazon AWS for empirical study. Their feasibility study revealed that FHE was good for cloud computing. Wang et al. [8] proposed usage of multiple keys for encryption.They proposed two schemes and found that performing operations on outsourced data is possible. Benzekki et al. [9] devised a methodology for HE usage effectively. Mohanta and Gountla [10] used FHE for cloud computing. Their scheme revealed the utility of FHE for flexible security. Li et al. [11] exercised on securing massive data in cloud platforms. General security provisions available for cloud computing are investigated by Hamdi [12]. Qin-long et al. [13] employed HE towards ensuring Digital Rights Management (DRM). It also considered privacy preserving method along with HE while [14] focused on application of HE in cloud and various security issues and encryption approaches are discussed in [15]. Ogbum et al. [16] illustrated an example for understanding usage of HE. Baharon et al. [17] proposed a lightweight HE for MCC. It was found good with approces like additive and multiplicative. Farokhi et al. [18] made experiments on the usage of semi-HELi et al. [19] defined a methodology to use HE for storage in cloud. The combination of RSA, MD5 and PHE are employed in [20] for both data security and integrity. HE is very useful encryption method as it allows search and other operations on the encrypted data as explored in [21]. The internal functionality of the HE is studied in [22] to see that encrypted data can be searched for. In [23] 2-DNF is explored for generating cipher texts. Data security is ensured with HE. The HE is employed for cloud computing in [24]. It was foud that there were issues with the traditional cryptographic methods due to key exchange problems. In order to overcome these issues, HE came into existence. HE allows flexible storage and compuations on encrypted data. The scheme is used to secure and outsource data prior to sending to cloud. Two categories of such schemes were found. They are known as Partially Homomorphic Encryption (PHE) which supports encryption and decryption with flexibility and Fully Homomorphic Encryption (FHE) that helps in encrypted storage and also facilitate search operations on the data without subjecting the encrypted data to decryption. AES with 128 bits is used and evaluated in [25] with respect to FHE for efficient search operations. Many HE schemes are found in [26]. HE has operations like encryption, decryption, key generation and evaluation. It also includes concepts like multiplicative HE and additive HE to enable different kinds of applications. Different HE schemes like Elgmal, Enhanced HE and Brakerski-Gentry-Vaikuntnathan encryption, to mention few, are explored in [27]. The usage of HE for both security and privacy is studied in [28]. An important example for FHE is known as Gentry [29]. In [30], [31] and [32], FHE is studied and found it to be ideal for dealing with security of cloud data and enable search operations[33][34]. There are many advantages of FHE. It supports search and data dynamics on outsourced encrypted data besides being flexible and useful for cloud storage. It is found in the literature that the FHE is used for unstructured data [35][36]. In this paper, we proposed a methodology that deals with structured, unstructured and semi-structured forms of data for secure storage and retrieval along with data dynamics.

3. SECURING DATA

Securing data when it is outsourced needs a standard approach. Even before storing data is public cloud, it has to be encrypted. As data is not maintained in the local devices, it is essential for data owner to secure it beforehand. Due to security concerns, the usage of cloud for storage may be deteriorated. The outsourced data needs to be maintained with data integrity. Cloud Service Provider (CSP) is not allowed to modify data. However, data integrity issues may arise due to many reasons such as misuse of data, stealing data and data leakage or damage due to hidden hardware issues. There might be other problems like external and internal attacks. Thus it is very crucialto have provision for security in both storage and retrieval of data.

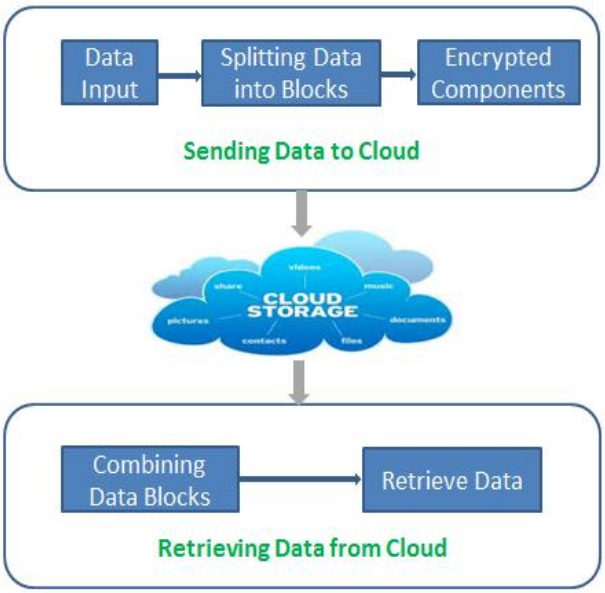

Figure 2: *Shows general procedure for secure storage and retrieval in cloud*

As shown in Figure 2, the input data is split into parts and then encrypted. The encrypted data is then send to public cloud. The encrypted data may be stored in different cloud servers. When data is retrieved, it needs to be decrypted

before it reaches end users. This phenomenon provides required steps to realize secure cloud storage and retrieval. Both data storage and retrieval are performed thus in a secure fashion.

4. PROPOSED FRAMEWORK

The traditional cryptographic methods have some limitations. Important limitation is that the encrypted data cannot be used directly for performing search and other operations. This is the major drawback. We proposed a framework for flexible and efficientstorage and retrieval of data. The framework also supports data dynamics on the encrypted data. Data owners need to store data with an encryption scheme but the encrypted data needs to allow search and other data modification operations without decrypting data. This is the first preference of the framework. Another important contribution of the framework is to support all the three kinds of formats of data. Unlike many other existing methods, the proposed method provides flexible data storage and retrieval besides data dynamics. In other words, it supports data manipulations and search over the encrypted data. Moreover, it helps in dealing with all kinds of data.

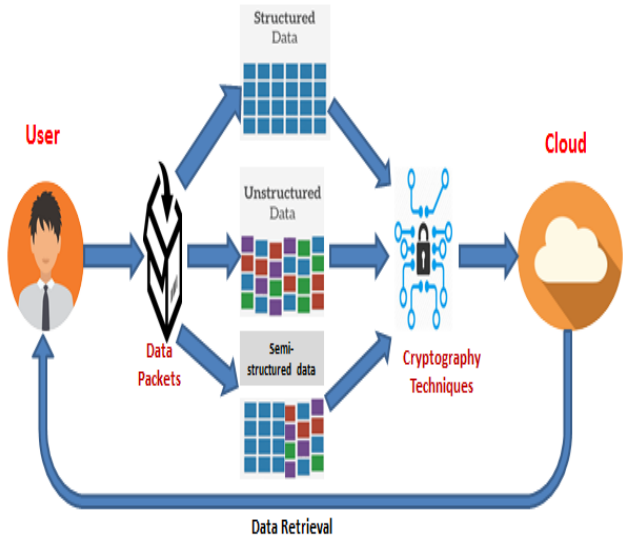

***Figure 3**:Overview of the framework for secure storage and retrieval*

According to the proposed framework provided in Figure 3, there is provision for dealing with structured, unstructured and semi-structured data. The cryptographic technique used in of FHE. Thus the outsourced data can support direct search and other data modification operations. This makes the framework flexible and efficient. An algorithm is proposed to support FHE. An application is built with intuitive interface to support the intended operations. The application supports connectivity with the public cloud known as Jelastic cloud. It interacts with MY SQL database (structured data) and MongoDB (unstructured and semi-structured data). As the application has underlying methods incorporated for FHE, end users can operate without the need for having much knowledge on cryptography and FHE. The framework enables data inputs to have secure storage. It can deal with all kinds of data. The framework exploits cryptographic primitives appropriately and achieves desired level of security and sophistication needed. The data heterogeneity is handled with the proposed framework. Based on the kind of data, the appropriate security measures are considered and the corresponding cloud database (MY SQL or MongoDB) is used in Jelastic cloud. Rules for RDBMS are provided by Dr. E. F. Codd. Such data is considered to be the data in structured format. This data is dealt with DDL and DML operations. When the given data is unstructured or semi-structured, then the FHE is employed on such data and data is maintained in the MongoDB.

4.1 Issues with the AES Based Encryption

When AES encryption is used for encryption of data stored in public cloud, it led to difficulties in searching on the encrypted data and also performing update and insert operations on it. The Algorithm 1 is used to know the tradeoffs between encryption security and also the flexibility in data dynamics.

```
Algorithm AESEncrypt()
{
        Analyse figure and round keys to know the
        arrangement of round keys
      Use plain text for state exhibit
        The beginning state cluster is updated with the
        round key
        State control is carried out with nine rounds
        Then carry out state control with tenth and the
        final round
        Use last state exhibit as scrambled cipher text
}
```

Algorithm 1:Encryption based on AESAES is a widely used cryptographic primitive. It works faster in encryption and decryption techniques. It is used to know the issues related search operations on the encrypted data. As shown in Figure 1, the given data is divided into number of pieces and they are subjected to encryption prior to outsourcing. When this algorithm is used for encryption, we found problems with the search and data dynamics in outsourced encrypted data. This is the reason that we preferred FHE based solution for flexible and efficient storage and retrieval where the data is searched in public cloud without actually decrypting. Algorithm 2 shows the proposed scheme towards cloud storage security.

```
Algorithm EFHomomorphicEncrypt ()
{
        Step 1: Two prime numbers denoted as p and q are
        selected
        Step 2: Arrived at the result by computing p and q
        N= p*q
        Step 3: GF(p) and arbitrary number denoted as x are
        selected where g<p and x<p
        Step 4: Compute and use y for encryption, gx mod p
        Step 5: Use two-step process for encryption
                a. Arbitrary whole number r is selected and
                   subjected to HE
                   E1(M) = (M+r*p) mod N.
                b. Arbiitrary whole number k is selected and
                   use computations for encryption as follows.
                   Eg(M) = (a, b) = (gk mod p, yk E1(M) mod
                   p)
        Step 6: Computations for decryption are carried out as
        follows.
        Dg() is M = b * (hatchet) -1 (mod p).
}
```

Algorithm 2: FHE based solution for data security

As provided in Algorithm 2, the given data is subjected to encryption. The encryption process is carried out in such a way that it allows search and other operations on the encrypted data that is outsourced to cloud platform. When data is of unstructured or semi-structured format, it is encrypted and saved to MongoDB, otherwise, it is encrypted and saved to MY SQL.

5. EXPERIMENTAL SETUP

Jelastic cloud is the famous cloud platform that provides Platform as a Service (PaaS) layer with multi-cloud support. It has a network of more than 60 data centres across the globe. It makes the storage services and application deployment services easier. It has the concept of creating environment for each user. For a cloud user, after due authentication, it allows creating environment needed for experiments. For the experiments of this paper, MY SQL is the environment chosen and the database is created. It also supports all application development platforms that can be configured. For instance, any environment related to Java, Ph. P, Python, Ruby, Node.js and so on.

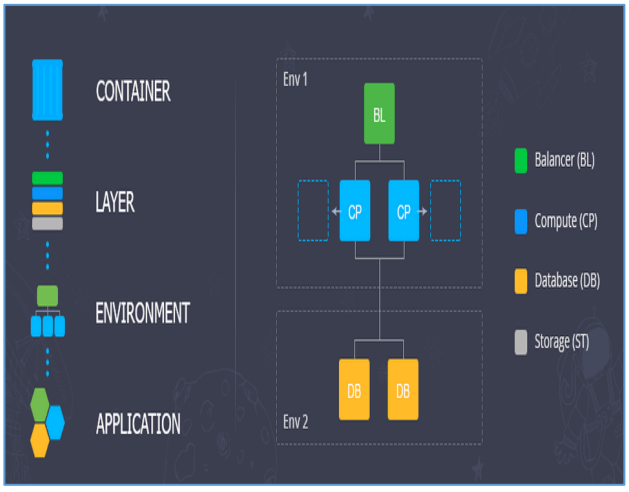

Figure 4: *Overview of Jelastic cloud components*

As presented in Figure 4, the Jelastic cloud platform has support for deploying different applications. Such applications need required environment. The environment can exist on top of different layers. There are containers to take care of the layers and environments. It has features like storage, database, compute and balancer. For this paper, EverData data center is chosen.

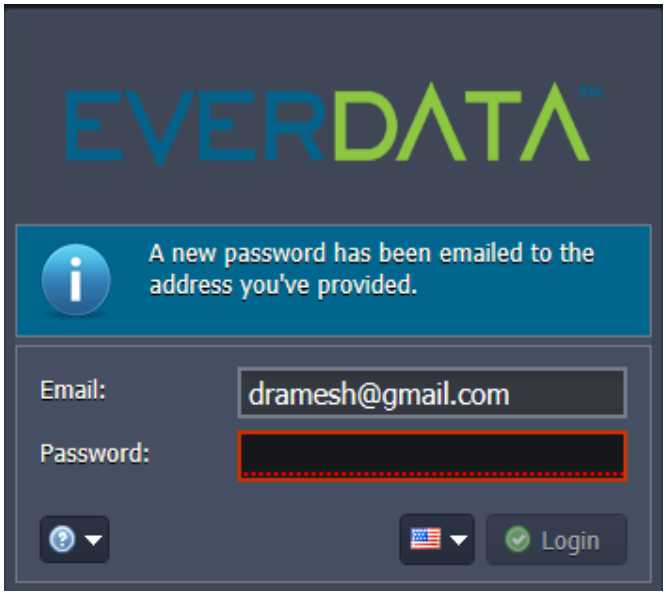

Figure 5: *Authentication process in Jelastic cloud*

As shown in Figure 5, the password is automatically mailed to the registered mail id. Then the password is used to get authenticated. Afterwards, the environment is set as shown in Figure 6.

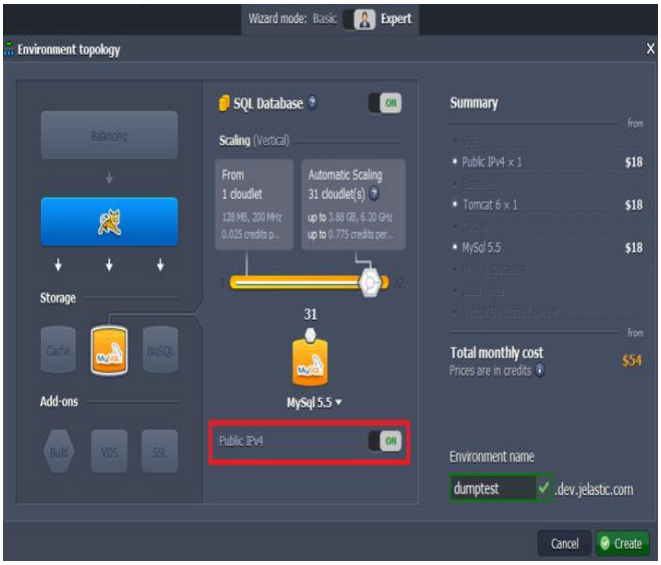

Figure 6: *Setting up MY SQL database environment in Jelastic cloud*

As presented in Figure 6, the MY SQL database environment is created. It allows remote database connectivity from any application. Java GUI application is built to connect and perform operations on the database. It

is related to structured data which is one of the data types of data.

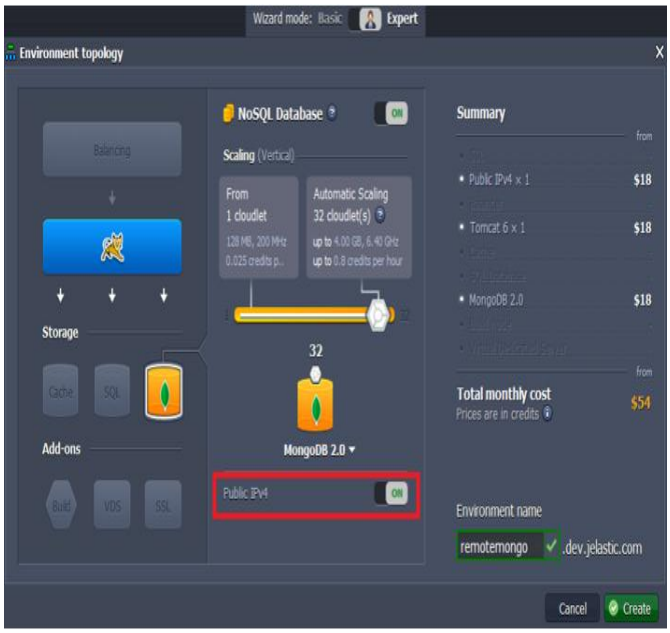

Figure 7:*MongoDb environment for NoSQL database*

As shown in Figure 7, with the configuration of MongoDB, it is possible to store and manipulate unstructured and semi-structured data types of data. In either environments, there is provision for public IP address that helps in reaching the cloud server and connect to required databases.

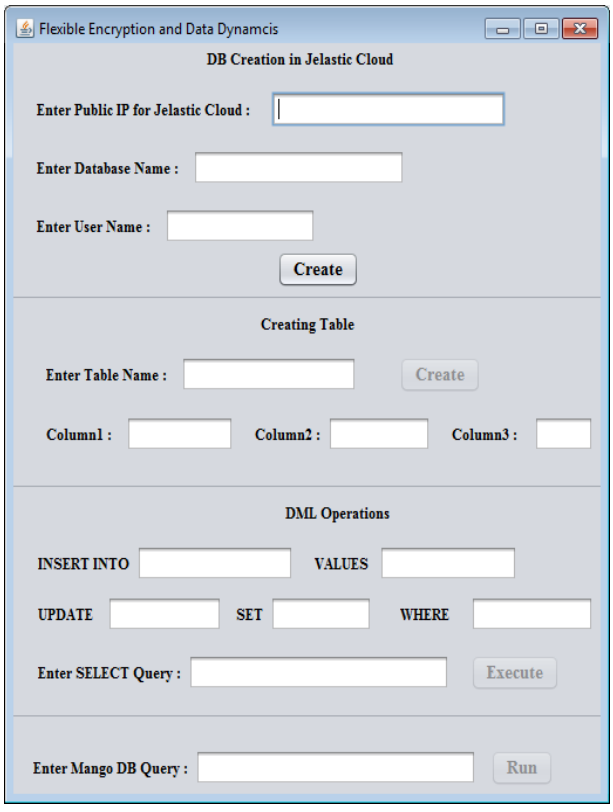

Figure 8: *Shows the prototype application to encrypt and outsource data and perform data dynamics*

As presented in Figure 8, the application built with GUI helps users to connect to remote cloud and databases like MY SQL and MongoDB. Besides it demonstrates the outsourcing of structured, unstructured and semi-structured formats of data. When data is submitted, it is encrypted and saved to public cloud. When queries are made to retrieve data, search operations are carried out without data being decrypted at server. When data manipulation is made, it also occurs on the encrypted data without decrypting it at the server. This way the proposed security scheme is flexible, efficient and supports data dynamics on encrypted cloud data.

6 EXPERIMENTAL RESULTS

The environment described in Section 4 is used to perform cloud based experiments on data storage security and flexibility in allowing operationson encrypted data. Towards this end, a prototype application is developed to have intuitive interface. Observations are provided in this section in the form of results of encryption, decryption and data dynamics besides total upload time, total execution time, encryption time and decryption time. The results showed the comparison between the proposed method and baseline AES method. Data with different size such as 10 MB, 50 MB, 100 MB and 500 MB is used for empirical study.

6.1 Difference in Execution Time

The proposed system is compared with the base line method in terms of execution time. The execution time is observed for both encryption and decryption operations when used with the application.

Data Size (MB)	Execution Time (sec)			
	Encryption Existing	Encryption Proposed	Decryption Existing	Decryption Proposed
10	0.8057	0.7989	0.7945	0.6921
50	2.5237	2.3956	1.9879	1.8925
100	2.7937	2.5968	2.5472	2.0156
500	13.6537	12.9896	9.6734	9.0132

Table 1: *Execution time comparison for encryption and decryption mechanismsAs shown in Table 1, the time taken by the proposed methodology and baseline method for encryption and decryption is presented against data sizes of data aforementioned.*

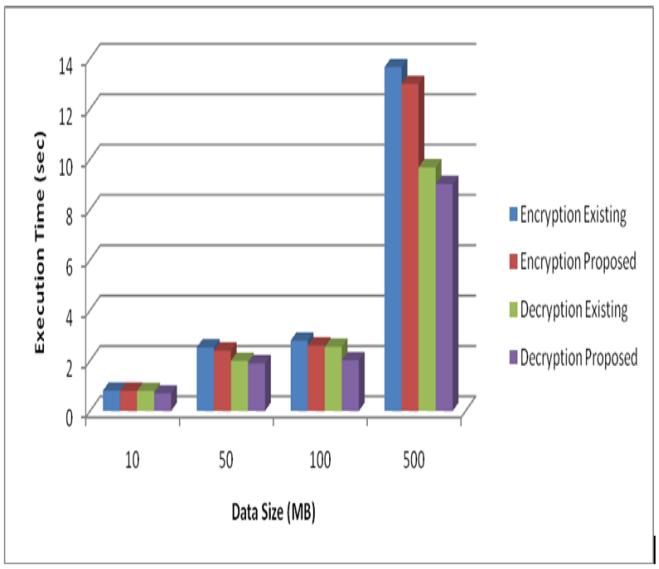

Figure 9:*Execution time comparison*

As presented in Figure 9, the size of data is presented in horizontal axis. The execution time is shown in Y axis. Two observations are found in results. The first one is that the size of data has its impact on the execution time. There is linear increase in the execution time as the size increases. The second trend is that the proposed system took relatively less time for encryption and decryption. These observations are for encryption and decryption times.

6.2 Time Taken for Upload

The total upload time is considered for proposed and existing schemes. The data size is changed and observations are made. The upload time includes the time taken for encryption of data and then outsource to public cloud. The time is computed in seconds and the data size is given in MB.

	Total Upload Time (sec)	
Data Size (MB)	Existing Scheme	Proposed Scheme
10	0.5862	0.4568
50	2.7162	2.1689
100	4.4762	3.3258
500	17.4262	13.9856

Table 2: *Total upload time of existing and proposed schemes against size of data*

As presented in Table 2, the total upload time required by the existing and the proposed systems for various sizes of dat are provided.

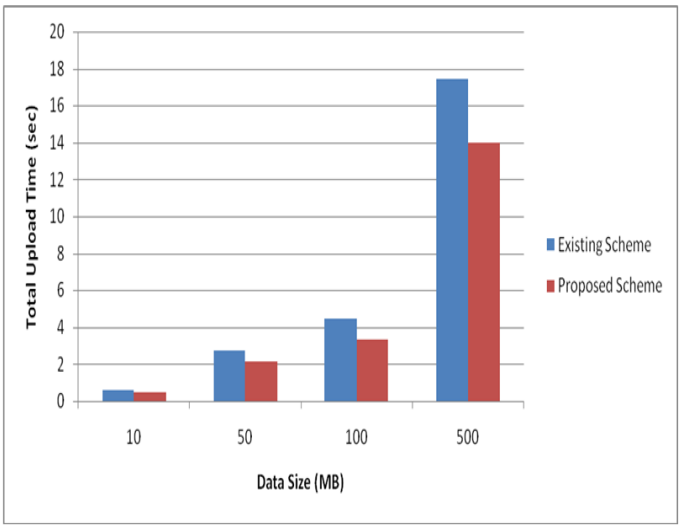

Figure 10: *Data size vs. execution time for total upload time*

As shown in Figure 10, the size of data is shown in X axis. It is given in MB while the vertical axis shows total upload time in seconds. The observations reevleaed that the volume of data has its impact on the total upload time. The results also reveal that the proopsed systm outperforms existing one.

6.3 Time Taken for Download

The download time of data with respect to existing and proposed methods is observed and compared in this section. The total download time is computed in seconds.

	Total Download Time (Sec)	
Data Size (MB)	Existing Scheme	Proposed Scheme
10	0.8058	0.7098
50	2.3237	1.9856
100	3.7937	3.4265
500	13.6537	12.3568

Table 3: Influence of the size of data on total download time

As presented in Table 3, the time taken for downloading data of different volumes for existing and proposed systems is provided.

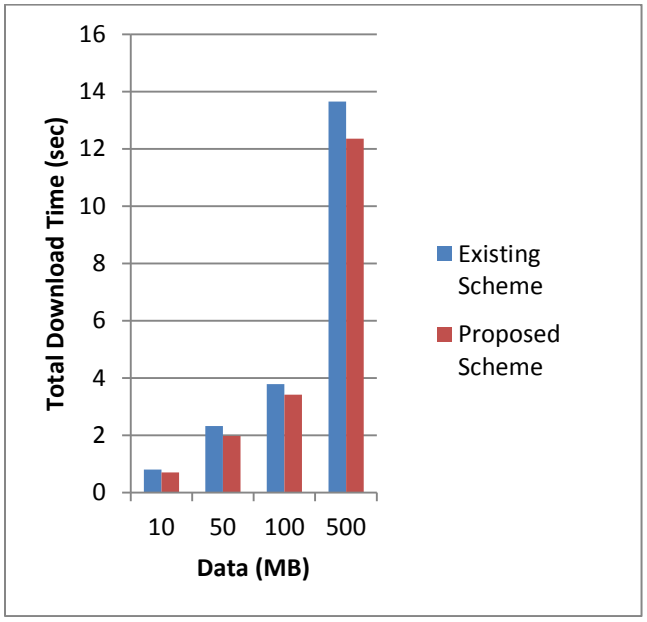

Figure 11:*Total download time against different volumes of data*

As presented in Figure 11, the time taken for downloading data is presented in vertical axis. The download time is against different volumes of data that is shown in horizontal axis. The size of data is found to have its impact on the total download time. At the same time, the proposed system outperformed the existing system with its performance. When it comes to the proposed encryption scheme, the observations are described here. Even when the given input is same, the proposed system showed uniqueness in choosing a random number and generating cipher text. When the number 10 is given as input initially, its random number is 1.1008E+154 and its cipher text is 846510095893784 2660898826910496 49270439150876. In the second experiment when the same number 10 is given as input, the corresponding r and cipher text values are 8.196E+153 and 240801595641083 597599840432248 452467630076927. This has proved that the system maintains uniqueness even when same data is given as input number of times. When it comes to decryption the cipher text 846510095893784 266089882691049 649270439150876 is converted into 10 and other cipher text 240801595641083 597599840432248 452467630076927 is also converted into original value 10. It concludes that there is consistency in the encryption and decryption procedures. When it comes to modifying an existing data, the application showed that it is possible without actually decrypting data at the server side. Search operation is carried out to perform search over encrypted data. Thus the proposed encryption scheme provides flexibility, efficiency and data dynamics on encrypted cloud data.

7. CONCLUSION AND FUTURE WORK

In this paper, we focused on the security of outsourced cloud data. When traditional encryption standard like AES is used, it needs decryption of data in the server before it is subjected to search and modifications. This is practically not viable due to the bulk of operations needed on the cloud and the magnitude of user base besides the unprecedented growth of requests to IaaS cloud for storage. To overcome the problem of traditional encryption mechanisms like AES for cloud computing, we proposed a framework that utilizes the proposed algorithm based on FHE. This algorithm encrypts data before outsourcing it to public cloud. The encrypted data is then used to perform search and modifications without being decrypted when queries are made runtime. In addition to this there are many advantages of the proposed system. 1) It supports all formats of data such as structured, semi-structured and unstructured. 2) It uses a real cloud environment with Jelastic cloud which is a multi-cloud IaaS platform. 3) It supports two environments in Jelastic cloud, one for MY SQL and one for MongoDB. In other words, it supports relational database for structured data and non-relational database for unstructured and semi-structured data. We have built a prototype application using Java SWING API which presents intuitive interface. It connects to Jelastic cloud over public IP address and allows end users to perform encryption, outsourcing, search, modification and downloading data without the need for knowledge on cryptography and the underlying mechanisms. In future, we intend to focus on the provision of security at the time of data publishing and data analytics in cloud computing.

REFERENCES

[1]. Zhao, F., Li, C., & Liu, C. F. (2014). A cloud computing security solution based on fully homomorphic encryption. 16th International Conference on Advanced Communication Technology. P1-4.

[2]. Maha TEBAA, Saïd EL HAJJI, Abdellatif EL GHAZI. (2012). Homomorphic Encryption Applied to the Cloud Computing Security. Proceedings of the World Congress on Engineering. 1, p1-4.

[3]. Maha TEBAA, 2Said EL HAJII. (2013). Secure Cloud Computing through Homomorphic Encryption. International Journal of Advancements in Computing Technology. 5 (16), p1-10.

[4]. Tebaa, M., Hajji, S. E., & Ghazi, A. E. (2012). Homomorphic encryption method applied to Cloud Computing. 2012 National Days of Network Security and Systems. P1-4.

[5]. Yu, J., Lu, P., Zhu, Y., Xue, G., & Li, M. (2013). Toward Secure Multikeyword Top-k Retrieval over Encrypted Cloud Data. IEEE Transactions on Dependable and Secure Computing, 10(4), 239–250.

[6]. Potey, M. M., Dhote, C. A., & Sharma, D. H. (2016). Homomorphic Encryption for Security of Cloud Data. Procedia Computer Science, 79, 175–181.

[7]. Kocabas, O., &Soyata, T. (2015). Utilizing Homomorphic Encryption to Implement Secure and Private Medical Cloud Computing. 2015 IEEE 8th International Conference on Cloud Computing. P1-8.

[8]. Boyang Wang, Ming Li, Chow, S. S. M., & Hui Li. (2013). Computing encrypted cloud data efficiently under multiple keys. 2013 IEEE Conference on Communications and Network Security (CNS). P1-5.

[9]. KamalBenzekki, Abdeslam El Fergougui and Abdelbaki El Belrhiti El Alaoui. (2016). A Secure Cloud Computing Architecture Using Homomorphic Encryption. International Journal of Advanced Computer Science and Applications. 7 (2), p1-6.

[10]. Bhabendu Kumar Mohanta, 2Debasis Gountia. (2013). Fully homomorphic encryption equating to cloud security: An approach. IOSR Journal of Computer Engineering. 9 (2), p46-50.

[11]. Li, Y., Gai, K., Qiu, L., Qiu, M., & Zhao, H. (2017). Intelligent cryptography approach for secure distributed big data storage in cloud computing. Information Sciences, 387, 103–115.

[12]. Hamdi, M. (2012). Security of cloud computing, storage, and networking. 2012 International Conference on Collaboration Technologies and Systems (CTS). P1-5.

[13]. HUANG, Q., MA, Z., YANG, Y., FU, J., & NIU, X. (2013). Secure and privacy-preserving DRM scheme using homomorphic encryption in cloud computing. The Journal of China Universities of Posts and Telecommunications, 20(6), 88–95.

[14]. Tebaa, M., Hajji, S. E., & Ghazi, A. E. (2012). Homomorphic encryption method applied to Cloud Computing. 2012 National Days of Network Security and Systems. P1-4.

[15]. Padmapriya. (2013). Cloud Computing: Security Challenges & Encryption Practices. International Journal of Advanced Research in Computer Science and Software Engineering. 3 (3), p1-5.

[16]. Ogburn, M., Turner, C., &Dahal, P. (2013). Homomorphic Encryption. Procedia Computer Science, 20, 502–509.

[17]. Baharon, M. R., Shi, Q., & Llewellyn-Jones, D. (2015). A New Lightweight Homomorphic Encryption Scheme for Mobile Cloud Computing. 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing. P1-8.

[18]. FarhadFarokhi, Iman Shames and NathanBatterham. (2013). Secure and Private Cloud-Based Control Using Semi-Homomorphic Encryption, p1-8.

[19]. Li, J., Chen, S., & Song, D. (2012). Security structure of cloud storage based on homomorphic encryption scheme. 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems. P1-4.

[20]. Ora, P., & Pal, P. R. (2015). Data security and integrity in cloud computing based on RSA partial homomorphic and MD5 cryptography. 2015 International Conference on Computer, Communication and Control (IC4). P1-6.

[21]. Shai Halevi, “Homomorphic Encryption Tutorial”, CRYPTO-2011, 1-60

[22]. Kristin Lauter, Michael Naehrig and Vinod Vaikuntanathan, “Can Homomorphic Encryption be Practical”, ACM, 2011, 1-18.

[23]. Dan Boneh, Eu-Jin Goh and Kobbi Nissim, “Evaluating 2-DNF Formulas on Ciphertexts”, 2006, 1-16 .

[24]. Ihsan Jabbar and Saad Najim, “Using Fully Homomorphic Encryption to Secure Cloud Computing”, Internet of Things and Cloud Computing. 4 (2), 2016, 13-18.

[25]. Craig Gentry, Shai Halevi and Nigel P. Smart, “Homomorphic Evaluation of the AES Circuit (Updated Implementation)”, 2015, 1-35.

[26]. Payal V. Parmar and Shraddha B. Padhar, “Survey of Various Homomorphic Encryption algorithms and Schemes”, International Journal of Computer Applications. 91 (8), 2014, 26-32.

[27]. Jean-Sebastien Coron, Avradip Mandal, David Naccache and Mehdi Tibouchi, 2011 “Fully Homomorphic Encryption over the Integers with Shorter Public Keys”, 2011, 1-24.

[28]. Maha TEBAA and Said EL HAJII, “Secure Cloud Computing through Homomorphic Encryption”, International Journal of Advancements in Computing Technology. 5 (16), 2013, 29-38.

[29]. Dan Boneh and John Mitchell, “A FULLY HOMOMORPHIC ENCRYPTION SCHEME”, 2009, 1-209.

[30]. Shashank Bajpai and Padmija Srivastava, “A Fully Homomorphic Encryption Implementation on Cloud Computing”, International Journal of Information & Computation Technology. 4 (8), 2014, 811-816.

[31]. Iram Ahmad and Archana Khandekar, “Homomorphic Encryption Method Applied to Cloud Computing”, International Journal of Information & Computation Technology. 4 (15), 2014, 1519-1530.

[32]. Kamal Benzekki, Abdeslam El Fergougui and Abdelbaki El Belrhiti El Alaoui, “A Secure Cloud Computing Architecture Using Homomorphic Encryption”, International Journal of Advanced Computer Science and Applications. 7 (2), 2016, 293-298.

[33]. D. Ramesh, B. Rama, “A Light Weight Cryptographic Technique for Secure Outsourcing and Retrieval of Data in Cloud Computing”, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8, Issue 10August 2019.

[34]. Venkateshwarlu Velde, B. Rama, “Optimized Weighted Round Robin Approach for Load Balancing in Cloud Computing” published in Journal of Computational and Theoretical NanoScience, 2019, Vol. 16, pp. 1-8, doi:10.1166/jctn.2019.7823.

[35]. Vurukonda, Naresh, and B. Thirumala Rao. "DC-MAABE: Data Centric Multi-Authority Attribute Based Encryption on Cloud Storage." Journal of Computational and Theoretical Nanoscience 16.5-6 (2019): 1893-1901.

[36]. Venkatakotireddy, G., B. Thirumala Rao, and Naresh Vurukonda. "A Review on Security Issue in Security Model of Cloud Computing Environment." Artificial Intelligence and Evolutionary Computations in Engineering Systems. Springer, Singapore, 2018. 207-212.