SECURED DATA STORAGE IN CLOUD USING HOMOMORPHIC ENCRYPTION

R Kanagavalli[1] and Vagdevi S[2]

[1]Department of Computer Science and Engineering, GAT, Bengaluru, India.
[2] Department of Electrical and Electronics Engineering, GSSSIETW, Mysuru, India.

ABSTRACT

Cloud data storage is a model of data storage where data in its digital form is stored in logical pools across the physical storage which is distributed across multiple servers usually in diverse locations. The full physical environment is owned and managed by an organization called Cloud Service Provider (CSP). In this type of storage, the data is distributed and owned by third party, and there is risk of unauthorized access. This makes security, reliability, confidentiality and privacy of data more important when storage in cloud is thought of. Although there are methods available for providing security, much work is not done with emphasis on dynamic nature of the data. Hence securing the data while updating becomes important. In this paper, a method that ensures confidentiality of data stored in cloud using homomorphic encryption is presented. The paper also provides a technique to ensure confidentiality during data updates in cloud. The presented method includes byte level automorphism for ensuring data integrity and confidentiality. The tabulated experimental results show that the proposed method provides more secure frame work for ensuring confidentiality and integrity of data in cloud.

KEYWORDS

Cloud Storage, Homomorphic Encryption, Privacy, Confidentiality & Data Integrity

1. INTRODUCTION

According to a definition given by the NIST (National Institute of Standards and Technology), cloud computing is defined as: “Cloud Computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”. NIST defined five essential characteristics that distinguish cloud from other technologies, namely: on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service [1][23][24].

Webster’s dictionary defines homomorphism as “a mapping of a mathematical set into or onto another set or itself in such a way that the result obtained by applying operations to elements of the first set is mapped onto the results obtained by applying those corresponding operations to their respective images in the second set”[2]. In Computer Science homomorphic encryption is used in the conversion of plain text to cipher text. Plain Text (PT) is any information like email messages, word files, images or credit card or cash transaction information which the sender wishes to transfer to a receiver. PTs are like an input to any algorithm which encrypts it into an unreadable form called Cipher Text (CT) .CTs is readable only when they are decrypted using key. Homomorphic encryption helps in this encryption –decryption process as follows: - it allows specific types of computations to be done on cipher text that produces an encrypted result which is again a cipher text. This cipher text is same as the outcome of the same operations computed on the plain text [3].

DOI: 10.5121/ijccsa.2019.9401

Data security in cloud depends on how a CSP ensures their clients in terms of Security, Reliability, Confidentiality, Liability and Privacy of their data. To provide security a method is proposed in this paper in which data to be stored in dispersed into many small parts before transmission regardless of the repository of the original data, and then homomorphic encryption is used to encrypt the data. It is assumed that all the operations both arithmetic and logical are represented in the form of logic circuits and are applied to the encrypted data. All the existing methods that have been studied concentrates on providing data security by considering the size of the public key, encryption time and decryption complexity while doing static operations. In real time applications the data is not static always but dynamic. So while concentrating on providing security in cloud, dynamic nature of the data should also be considered. The rest of the paper is organized as follows. Section 2 presents a brief summary of the literature study that was done to carry out the research. Section 3 describes the proposed method while section 4 provides the necessary results to substantiate the outcomes. Section 5 summarizes the study and the future directions of work.

2. RELATED WORK

From the time the concept of privacy homomorphism was introduced in 1978, homomorphic encryption is in the minds of the researchers like a holy grail and the search somewhat ended in with the proposal of a homomorphic encryption scheme by Gentry in 2009. This is the first homomorphic encryption scheme that supports both addition and multiplication on encrypted messages. Here the number of permissible homomorphic operations is unlimited [5]. Consider an example which consists of a simple encryption function as: if *x* is a plain text, then the cipher text *y*=Enc(*x*) is given by *y*=3*x* and decryption is given by Dec(*y*) =*x*/3. If homomomorphic technique is followed then simply by applying the operation on the cipher text will yield the same as applying the operation to the plain text after downloading and decrypting. The below Figure 1 shows an example .Consider if the plain text is 3 and 5 respectively and using the encryption function described above the corresponding cipher texts become 9 and 15 respectively. If the operation applied is addition, then the addition of 9 and 15is 24 which when decrypted yields the same result when 3 and 5 are added i.e. 8. Thus by using homomorphic encryption the necessary operations can be done on cipher text itself without downloading and decryption.

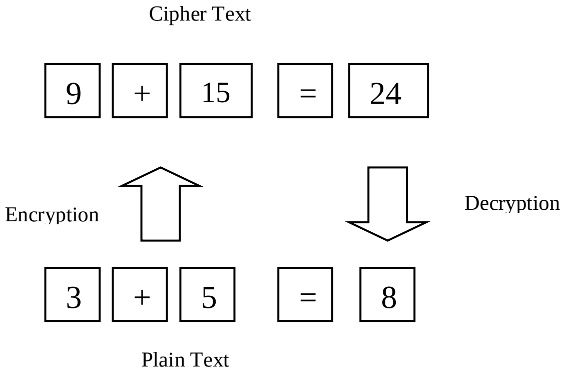

Figure 1. Homomporphic encryption operation applied to numbers as plain text

Homomorphic encryption can also be applied to string texts also which follows the ideology similar to Ceaser Cipher Substitution method [4]. The below Figure 2 shows an example of the method for concatenation operation.

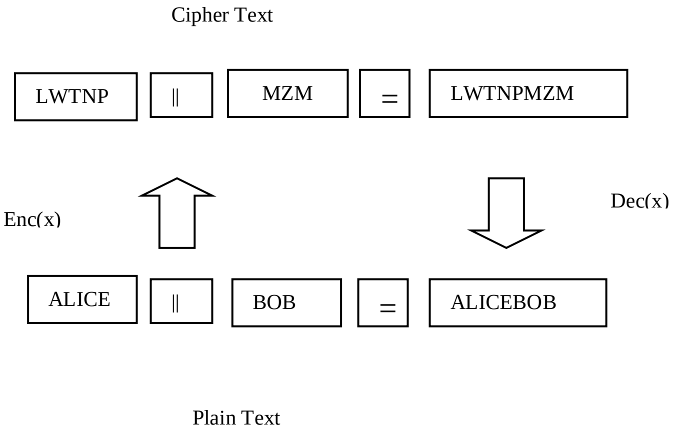

Figure 2. Homomporphic encryption operation applied to strings as plain text

In [6], Gentry-Halevi proposed a lattice based homomorphic encryption scheme in which the refresh procedure was enhanced when compared to original scheme. In [7], the authors have proposed a method which applies “squash decryption” to get a boot strappable fully homomorphic scheme. In [8], the authors proposed a method in which only a small subset of the public key is stored. In [9], the authors suggested an improved scheme to overcome heuristic attacks on integers. Here the secret key is taken as a matrix and to implement FHE the cipher texts of the secret key and public key are added. In [10], authors suggested a method in which re - linearization technique was used. Here cipher text is shortened and decryption complexity is reduced. In [11], the authors have suggested a FHE scheme called as modulus switching with linear efficiency whereas the modulus size is sacrificed. The following Table 1 gives a summary of the literature study made.

Table 1. Survey of Existing Methods

Algorithm → **Characteristics ↓**	**Paillier**	**RSA**	**El-Gamal**	**Gentry**	**BGV**	**AHEE**
Cloud Application	Yes	Yes	Yes	Yes	Yes	Yes
Homomorphic Encryption Type	Additive	Multipli-cative	Multipli-cative	Mixed	Mixed	Mixed
Keys used	Two	Two	Two	Two	Two	Two
Security Applied to	CSP	CSP	CSP	CSP	CSP	CSP
Nature of Data	Static	Static	Static	Static	Static and dynamic	Static and dynamic

From the study of existing literature, it is clear that there are few methods available addressing the dynamic nature of data but not ensuring all the security aspects. In the following section the model that addresses the security considerations with dynamicity is proposed.

3. WORK DONE

A typical cloud data storage model consists of three main components namely i) User ii) Cloud Service Provider (CSP) and iii) Third Party Auditor (TPA) and is as shown in Figure 3 .

i) User: A person who creates, stores, and uses the data may be personal or organizational by utilizing the services provided by CSP.

ii) Cloud Service Provider (CSP): An Organization who manages the cloud server and provides services to the user as per measured service policy.

iii) Third Party Auditor (TPA): An optional component also known as verifier. If the user is suffering from lack of timing, then the data verification is done by TPA.

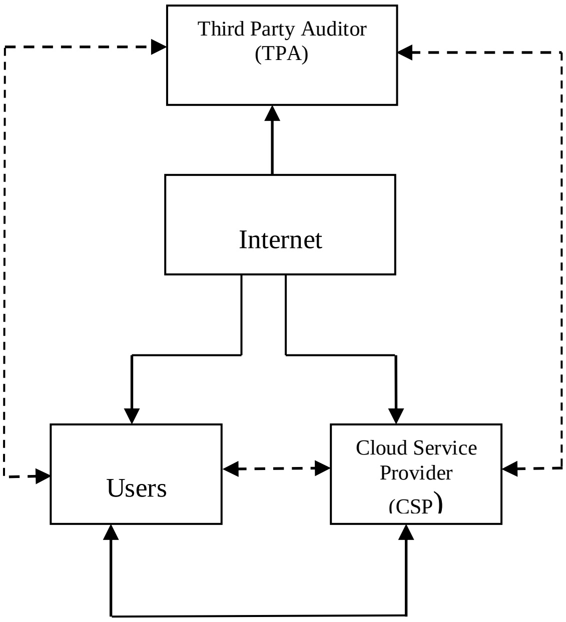

Figure 3. Cloud Data Storage Model

There are two types of security threats that may arise in a typical cloud storage system. Internal Threats are threats internal to the cloud system. Here, the CSP can leak the information of the user or may use it for own by modifying the data. External Threats are the threats caused by some external agents or outside third party. In this of attack, stored data is misused for some unjustified work. To overcome these issues, in the proposed work the data is dispersed into many small parts regardless of the repository of the original data. Our aim is to cover two security issues namely confidentiality and privacy of the data stored and to achieve our aim homomorphic encryption methods are used as they are highly capable of providing data security in terms of confidentiality and privacy. The proposed method is built upon considering BGV method as the basis [17] method and the basic functions of BGV method are as shown in Figure 4.

Encrypt (P m, PU pub): C ct
Decrypt (C c, PR priv): P m
Level shifting operations Rescale (C ct): C ct’ Switch Key (Augmented C ct): C ct’
Homomorphic operations Add (C c1, C c2): C csum Mul (C c1, C c2): C cmul

Figure 4. BGV Encryption Scheme

In this work attempt is made to enhance the efficiency of BGV method by modifying the key switching matrix with the help of permutation and the metadata production is computed based on Elliptic Curve Cryptography method. The methodology of the proposed method is explained as follows.

Let **F** be the file to be stored in cloud. The file is divided into **T** blocks of equal length ***n***. Every block **T** will be applied with FHE algorithm. The switching matrix is defined in terms of permutation function which is calculated as follows

Let f (key) be a random function indexed on some key k and the function f is defined as

$$f \leftarrow \{0,1\}^{*} X\, key \{0,1\}\ \log_2 n^2 \qquad (1)$$

The permutation function is defined as

$$\Pi(key) \leftarrow \{0,1\}^{*} \log_2 n\ X\, key \{0,1\}^{*}\ \log_2 n^2 \qquad (2)$$

In the methodology there are 3 procedures used namely Key Production , Encryption and Meta data production which are explained as follows

3.1. Key Production

The steps followed in the Key Production procedure are given below.
Step 1. Choose two keys private (PR) and public (PK) with length ***k*** where ***k***>512.
Step 2. Choose two large prime numbers ***p*** and ***q*** whose size is ***k*** and ***p≡q≡2***.
Step 3. Let N_n be the order of the switching matrix over the ring Z_n.
Step 4. Let *y* be the randomly chosen integer such that *gcd(y, n) =1*.
Step 5. Consider N_n as the generator of the essential key pairs.
Once the necessary keys are generated, the data is encrypted as per the following encryption procedure.

3.2. Encryption

The steps followed in the Encryption procedure are given below.
Step 1. The data file **F** is divided into **T** blocks of equal length and let ***b*** inputs are there in **F**.
Step 2. Each block t_i is encrypted with the switching matrix generated in the previous procedure.
Step 3. Now the file **F** can be written as $F=\{t_1,t_2,t_3,...t_N\}=\{t_i\}, 1 \le i \le N$.
Step 4. Now The encrypted file **F'** can be written as $F' = \{t_i'\} = t_i + \Pi(b)$.
The execution of encryption procedure increases the confidentiality of data.

3.3. Meta Data Production

Once the encryption is done the user computes Meta data over the encrypted data in order to ensure the privacy of data. The metadata computation is done as follows:
Step 1. The encrypted data t_i' the public key PK and the private key PR are taken as input.
Step 2. The metadata block M_i is computed as $M_i \leftarrow t_i' * P \pmod{N_n}$ where $P \in E_n(0, b)$.
Step 3. The metadata block M_i is produced as output.

This step ensures the privacy of data. Once the Meta data is calculated, the user sends the file **F'** to cloud server for storage and keeps the Meta data for later verification.

3.4. Query Method

This procedure Query and the following Reply procedure are used for testing the model with the challenges that may arise during deployment and are also used for proving that the proposed model is suitable for providing confidentiality and integrity of data.

In this method, the challenger takes two random keys say ***key*₁** and ***key*₂** as input and computes *f* **(*key*₁)** and *f* **(*key*₂)**. Then a block of data is chosen by computing the random indices using Π function. This avoids the foreseeing of server for block identification. Next a query is created a triple of the form (K, j, W) where K= {*f* **(*key*₁)**, f **(*key*₂)**}; j is the block and 1≤j≤N and W is the data to be queried. Once the query is created it is sent to server for reply.

3.5. Reply Method

When the server receives the query, a response R is generated by taking the encrypted data and query as input. R is computed as follows:

Step 1. Generate a sequence of n random numbers.
Step 2. Calculate y from the set of random numbers as

$$y = \sum_{j=1}^{n} a_j\, m_j \qquad (3)$$

Step 3. Calculate R as R= y. (K, j, W) (mod n)

The integrity of data is checked as given below

Let R' be the data stored orignially and let R be the data generated as reply to the query.

To prove that $R' = R_j$

$$WKT\ R' = y\, S(mod\ n)\ where\ S\ is\ the\ query \qquad (4)$$

$$Also\ S = \prod_{j=1}^{n} 1. a_j\ \ t_i\ \ where\ a_j\ = f(j) \qquad (5)$$

$$= \prod_{j=1}^{n} a_j\ (P\ mod\ N_{\ n}\ )mod\ n \qquad (6)$$

$$\therefore\quad R' = y\left(\ \prod_{j=1}^{n} a_j\ (P\ mod\ N_{\ n}\right) mod\ n \qquad (7)$$

$$= R_j \qquad (8)$$

Since R’=R_j where j is the number of block which is chosen at random, it is clear that data stored is intact.

4. RESULTS AND DISCUSSION

The proposed scheme has been tested for text files of different sizes and the results are compared with RSA algorithm which is the homomorphic encryption method using one operation and AHEE[19] algorithm which is mixed homomorphic encryption on algebraic polynomials. The parameters tested are size of cipher text, encryption time and decryption time.

The below Table 2 shows the size of the cipher text (in bits) generated for various methods and the same is represented in the graph as shown in Figure. 5.

Table 2. Size of Cipher Text

Size of PT	RSA	AHEE	Proposed FHE
24	30	28	26
72	80	78	76
128	137	133	133
256	272	263	263
1024	1033	1027	1027
2048	2057	2055	2053
4096	4123	4115	4107
8248	8313	8295	8273

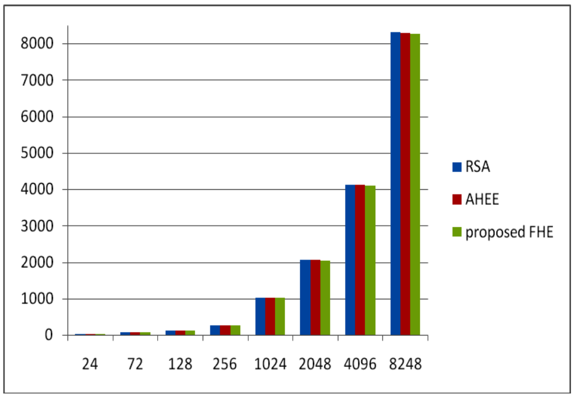

Figure 5. Size of Generated Cipher Texts

The below table 3shows the time taken for encryption (seconds)for different files with sizes from from 24 bits(3bytes) to 8248 bits(1031 bytes) and the Figure 6 depicts the graph plotted for the same.

Table 3. Encryption Time

Size of PT	RSA	AHEE	Proposed FHE
24	12.05	5.67	4.13
72	36.15	17	12.39
128	64.22	30.2	22.03
256	128.46	60.5	44.05
1024	514.05	242	176.21
2048	1024.25	477	352.43
4096	2056.45	968	704.85
8248	4133.15	1949	1419.34

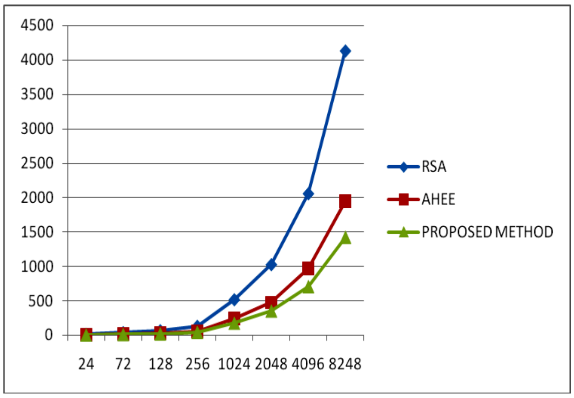

Figure 6. Time Taken for Encryption

The following table shows the decryption time (seconds) of different files with size varying from 24 bits (3 bytes) to 8248 bits (1031 bytes) and the same is represented in the graph as shown in Figure 7.

Table 4. Decryption Time

Size of PT	RSA	AHEE	Proposed FHE
24	5.96	0.92	0.92
72	17.88	2.76	2.76
128	31.78	4.9	4.75
256	63.57	9.8	9.25
1024	254.31	39.3	37.13
2048	508.59	78.5	76.23
4096	1017.17	157	155.87
8248	2048.27	316	312.12

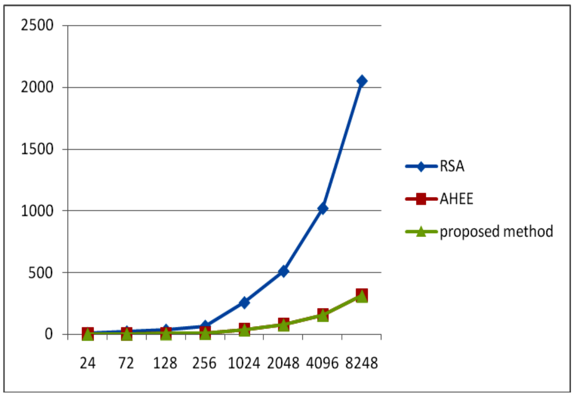

Figure 7. Time taken for Decryption

The proposed method prevents the data leakage through eavesdropping because of the query function . Here the secret parameter is chosen at random by the user and hence any combatant will not be able to dig any data from the encrypetd file F'. The below table and figure shows the key generation time and verification time for files with size from 16 bytes to 1031 bytes with fixed block size.

Table 5. Time Taken for Key generation and Verification

File size (bytes)	Key generation time (sec)	Verification time (sec)
16	1.42	1.5
32	2.84	3.12
128	11.36	12.42
256	22.72	24.15
512	45.44	48.33
1031	91.5	96.65

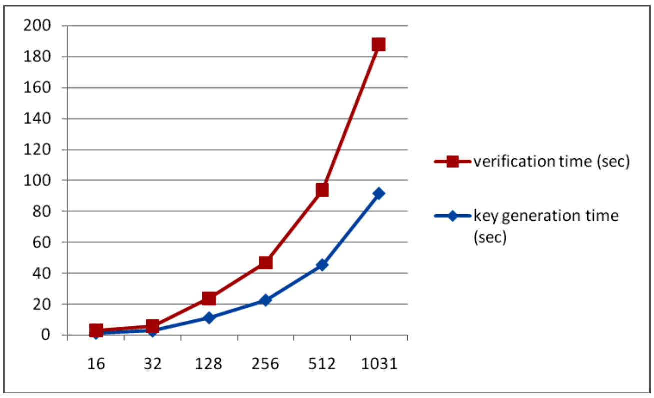

Figure 8. Key Generation and Verification Times

From the figures 6and 7 it is clear that the propsed method is taking less time compared to the other methods for encryption and decryption .Also from figure 5 it is shown that the size of the cipher text by the propsed scheme is smaller. The figure 8 depicts the key generation and verfication time required to query a data block which substanitaes the equation 8. It also proves that as the size of the file increases , the verification time also increases and hence integrity of data is ensured.

5. Conclusion

In this paper, a method that solves the problem of data confidentiality and privacy in the cloud data storage system is proposed. The proposed method uses a fully mixed homomorphic encryption technique with byte level automorphism based on BGV scheme.. The presented results shows that the proposed work provides better security parameters when compared to existing methods. The present study is made by concentrating on insertion, deletion and updation operation with fixed block size of files. In our future work we are planning to work in the direction of improving the scheme with variable block sizes.

REFERENCES

[1] P. Mell, T. Grance, “The NIST Definition of Cloud Computing,” National Institute of Standards and Technology, U. S. Department of Commerce, September 2011.

[2] The Miriam-Webster’s Dictionary website.[Online]. Available: http://www.merriam-webster.com

[3] B.Hayes,”Alice and Bob in Cipherspace” in American Scientist, vol.100, 2012, paper 5, p.362.

[4] Monique Ogburn, Claude Turner, Pushkar Dalal,”Homomorphic Encyption”, Procedia Computer Science 20(2013),pp.502-509.

[5] Craig Gentry,”A fully homomorphic encryption scheme”, PhD thesis, Stanford University, 2009.

[6] C.Gentry and S.Halevi,” Implementing gentry’s fully homomorphic encryption scheme”, EUROCRYPT, 2011, pp.129-148.

[7] M. van Dijk, C. Gentry, S. Halevi and V. Vaikuntanathan, Fully homomorphic encryption over the integers. In H. Gilbert (Ed.), EUROCRYPT 2010, LNCS, vol. 6110, Springer, 2010, pp. 24-43.

[8] J.S.Coron, T.Lepoint and M.Tibouchi,”Batch Fully homomorphic encryption over the integers”, IACR Cryptology ePrint Archive, 2013.

[9] G.ChunSheng,”Attack on Fully Homomorphic Encryption over the Integers”, arXiv: 1202.3321,2012.

[10] Zvika Brakerski and Vinod Vaikuntanathan,”Fully homomorphic encryption from ring-LWE and security for key dependent messages”, CRYPTO 2011, Vol.6841, 2011, pp.501-510.

[11] Iram Ahmad,Archana Khandekar,”Homomorphic Encryption Method Applied to Cloud Computing”,International Journal of Information &Computation Technology,Vol.4,2014,pp. 1519-1530.

[12] Khalid El Makkaoui,Abdellah Ezzati,Abderrahim-Beni-Hssane,Cina Motamed,”Data Confidentiality in the World of Cloud”,Journal of Theoretical and Applied Information Technology, Vol.84,No.3,2016,pp.305-314.

[13] Majedah Alkharji,HangLiu,MayyadaAlHammoshi,”A Comprehensive Study of Fully Homomorphic Encryption Schemes”,International Journal of Advancements in Computing Technology,Vol.10,No.1,2018,pp.1-24.

[14] Caihui Lan, Haifeng Li,Shoulin Yin,Lin Teng,”A New Security Cloud Storage Data Encryption Scheme Based on Identity Proxy Re- encryption”,International Journal of Network Security,Vol.19,No.5,2017,pp.804-810.

[15] Khalid El Makkaoui, Abdellah Ezzati, Abderrahim-Beni-Hssane,”Cloud-ElGaml: An Efficient Homomorphic Encryption Scheme”, IEEE, 2016, pp.1-4.

[16] Nasarul Islam K V, Mohamed Riyas K V,"Analysis of Various Encryption Algorithms in Cloud Computing”, International Journal of Computer Science and Mobile Computing, ISSN: 2320-088X, 2017, pp.90-97.

[17] R.Kanagavlli ,Dr.Vagdevi S ,” A Mixed Homomorphic Encryption Schemes for Secure Data Storage in Cloud”,IEEE International Advanced Computing Conference ,IACC2015,2015,D:O:I: 10.1109/IACC.2015.7154867.

[18] Bhagyashri R Hanji, Rajashree Shettar,” Stable Reduced Link Break Routing Technique in Mobile AdHoc Network”, ICACDS2018, Springer series, 2018, pp.74-83.

[19] Xiang Guangli,Cui Zhuxiao,"The Algebra Homomorphic Encryption Scheme Based on Fermat's Little Theorem",International Conference on Communication Systems and Network Technologies(CSNT),2012,pp.978-981.

[20] Sumitha J ,S.Manjupriya,"Comparative Analysis of Homomorphic Encryption in Cloud Computing",International Journal of Management,Technology and Engineering,Vol.8 ,No12,2018,pp.1251-1255.

[21] R.Kanagavlli, Dr.Vagdevi S," A Survey of Homomorphic Encryption Schemes in Cloud Data Storage,"International Journal of Recent Development inEnginerring and Technology, Vol.3, No.1, 2014, pp.71-75.

[22] R.Kanagavalli, Dr.Vagdevi S," Comparative Study of Homomorphic Encryption Methods for Secured Data Operations in Cloud Computing", ICEECCOT 2017, IEEE, 2017, D: O:I:10.1109/ICEECCOT 2017.8284566.

[23] Kashif Munir,Dr.Sellappan Palaniappan,"Framework for Secure Cloud Computing", International Journal of Cloud Computing :Services and Architecture,Vol .3,No.2,2013,pp.21-35.

[24] Nabeel Zanoon,"Toward Cloud Computing:Security and Performance", International Journal of Cloud Computing :Services and Architecture,Vol.5,No.5,2016,pp.17-26.

AUTHORS

R.Kanagavalli is working as Associate Professor in Department of Computer Science and Engineering, Global Academy of Technology, Bangalore, India. Her research interests include Cloud Computing, Artificial Intelligence and Data Analytics.

Dr.Vagdevi S is working as Professor and Dean, GSSSIETW, Mysuru, India. She has published many scholarly articles in journals and conferences. Her research interests include Network and Information Security, Cloud Computing, and Big Data Analytics.