International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-10, August 2019

Ensure Data Protection in Cloud Computing using ASCII Based Encryption Technique

Shivanna K, Prabhudeva S

Abstract*: *Cloud computing is an engineering discipline which is concerned with all aspect of storing and sharing useful data. Today, millions of users are able to upload and download useful information on cloud environment from anywhere at any time. Recently, many security issues are encountered in real time applications with a great loss of data and information. Subsequently, many research works have been done to highjack the security issues of cloud computing. Still, all feasible solutions may not applicable at all time and some practical situations. In this paper, we designed and implemented an ASCII based encryption technique that securely uploads, sharing the data on cloud framework with a minimum computational overhead and entropy. The proposed technique will increase the throughput of the system and it can be deployed on a cloud environment for sharing and securing useful data information.

***Index Terms*: Cloud Storage, Confidentiality, Encryption, One Time Secret Key**

I. INTRODUCTION

As long as, an increased computing and communication network, huge data has been generated, so greater storage space is required. Once the data is uploaded, we have to take care about its security, because client who stores the data that not have a control over it, this leads to security risks in terms of confidentiality, originality and availability of data and their services. Confidentiality has plays a major roll among other threats, since security will fails then the entire system will compromise. On the other hand, vast amount of data has been processed on cloud platform; we take care about execution cost. Since, the data security is the top most issues related to the cloud storage, we intentionally look forward to design best possible solutions that provides affordable services within a reachable cost. Based on the various encryption techniques and their computational cost, we will have to study low cost encryption technique. The security is applied based on type of data and their values. For example, securing credit card number and its pin number is different from securing general message shared between end users. Now we consider resources of cryptosystem that helps to provide best possible services to end users within a affordable cost.

In this paper we proposed a symmetric encryption of plain text using ASCII value. The ciphertext symbols generated using ASCII values depends on message length, random function and some matrix operations. Thus, the ciphertext generated by the proposed method will be unpredictable and output not depends on any specific key. So the prediction or cracking of data by the intruder will be difficult for the intruder.

The paper is structured as follows: In section II, existing work related to cloud security and what are the cryptographic techniques for solving security issues are highlighted. Section III provides design consideration of proposed method for data protection in cloud computing. Section IV summarizes experimental results. Section V provides security measures of proposed system. In section VI, we concluded this paper.

A. Motivation

In a cloud environment, most of clients upload their data on cloud service provider (CSP) and check their confidentiality of data in any geographical location. Any confidentiality violation leads to major loss and availability of data. Following are security problems identified based on earlier literature: 1. Data has been outsourced is not controlled by the user. 2. Cloud service provider (CSP) would face the key management issues. 3. Some CSP's does not have knowledge for encryption of data that may be stored in as original form. 4. Most importantly, we compromised on speed of encryption and decryption.

II. RELATED WORK

In [1], the researchers have proposed a symmetric encryption algorithm for securing data on cloud storage platform. They dealt with new encryption technique that converts original data into ASCII code and demonstrate the use of two symmetric keys for encryption and decryption. In this paper, researchers are unable to justify why two symmetric keys supposed to use for encryption and decryption?. In paper [2], a survey on the cryptographic encryption mechanisms were addressed based on the encryption and decryption time, throughput, memory, avalanche effects, correlation assessment and entropy. They suggested that Blowfish algorithm is the best option for various applications with respect to memory and encryption/decryption time and it is efficient in software. Their future work includes, requirement of hybrid encryption algorithms based on the necessary parameters that are used to enhance the overall security of encryption techniques. Sultan Aldossary et al. [3] summarized the data security, privacy, availability and integrity issues and their current solutions.

Manuscript published on 30 August 2019.
***Correspondence Author(s)
Shivanna K, Research Scholar, VTU, Belagavi and Asst. Professor, Dept. of CSE, GMIT, Davanagere, India.
Dr. Prabhudeva S, Information Science and Engineering, J N N College of Engineering, Shimoga, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license http://creativecommons.org/licenses/by-nc-nd/4.0/

They described the data storage in a remote server leads to some security issues, availability of data if it is required and possible solutions that protect data by cloud service provider while it is shared among many users.

Bin feng et al. [4] proposed an efficient protocol with bidirectional verification for storage security in cloud computing. They utilized a new entity to generate the authority credentials. So that no longer to assume that every TPA is credible. But this scheme may lead to have higher computational load at a higher security level whenever large file has been uploaded. Abha sachdex et al. [5] proposed AES algorithm to enhance cloud computing security. With the tremendous improvement of sensitive information on cloud, they suggested the need of security on sensitive information. Before data has been launched in the cloud, data have been encrypted using Advance Encryption Standard (AES). Since longer key length (128 bits) used by AES, it takes more time to search a key for encryption and decryption. Review of data security algorithm described in [6][7] [8]. They pointed out open issues and requirements of new technology for solving the data security issues. Mazhar Ali et al. [9] proposed secure data sharing in cloud. The proposed technique provides data confidentiality, secure data sharing without re-encryption. The encryption and decryption are done by Cryptographic Server (CS). Priyadarshini Patil et al. [10] done the comprehensive evaluation of cryptographic algorithms such as DES, 3DES, AES, RSA and Blowfish. They suggested that a user needs a cryptographic algorithm which is of low cost and high performance and analyzed the performance of cryptographic algorithms.

Faiqa Maqsood et al. [11] presented a comparative analysis for modern cryptographic techniques. The researchers have evaluated the performance of different symmetric and asymmetric algorithms by considering multiple parameters such as encryption/decryption time, key generation time and file size. They compared simulation results of different algorithms that clearly depict which algorithm is most suitable while achieving a particular quality attribute. The performance results show that the symmetric schemes are computationally inexpensive when compared with asymmetric schemes. Mansoor Ebrahim et al. [12] presented a comparative analysis of different existing symmetric algorithms based on their architecture, scalability, flexibility, reliability, security and limitation that are essential for secure communication. They observed that AES was the best among all in terms of security, flexibility, memory usage, and encryption performance. Muhammad Faheem Mushtaq et al. [13] proposed a key generation technique based on triangular coordinate extraction for hybrid cubes. They presented a key generation technique based on TCE for HC of order 4 matrices, which can be used to generate the keys during rotation of HCs. Design of key generation technique, is the major concern for security. In this technique, the four key matrices are employed in the generation of one TKM matrix using the concept of TCE. The modulo-16 operation used to calculate the value of TKM by utilizing the rotation points and then the rotation of HCs increases the complexity in the design of RCM. It creates a difficulty to predict the pattern of keys from cryptanalysts.

Mohd Rizuan Baharon et al. [14] proposed a improved homomorphic encryption scheme for cloud computing. They uses a symmetric key for encryption along with a protocol to implement the scheme. They also provide an analysis that related to the noise growth in the processed ciphertext and squashing technique which is required to reduce the noise. The analysis might be improving the efficiency of the scheme as the squashing technique is time consuming. Kim-Kwang Raymond Choo et al. [15] presented potential topics for future research work in cloud security engineering. They include advanced security features, cloud-based intrusion detection and prevention systems, distributed authentication and authentication, implementation of cryptographic and key management strategies in clouds (e.g. homomorphic encryption for cloud computing) and security-focused service level agreements, cloud auditing and certification.

III. PROPOSED METHOD

The proposed method is designed to provide a secure solution to the data shared online. The notation throughout the paper is summarized in Table I.

Table I: Notations and descriptions.

Notations	Descriptions
M	Message
L_t	Length of the message
V_i	ASCII value of message
P_n	Prime number
N_p	Non-prime number
Mat_i	Matrix formed by [P_n || N_p]
Mat_j	Matrix formed by adding sk
C	Ciphertext
D	Decryption
SK	Secret Key

A. System Model

In our scheme, we subdivide all entities as illustrated in Fig. 1 (the arrow represent flow of data, the rectangle represents entities used in this paper).

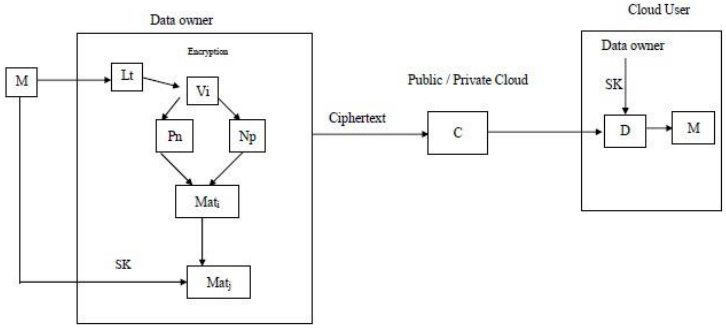

Fig. 1: System Architecture

Data owner, as the major entity of this scheme has to store information on cloud storage Fig.1. The CSP, acts as a server of this system provides storage services and responsible for all type of services to its users. Cloud user can access the original data by using one time secret key stored by data owner.

© Copyright: All rights reserved.

This means that only authorized person can access the data without any interruption. In the encryption process the plain text is first converted into ASCII value as discussed in the research paper [16], the first step is to generate ASCII value (Algorithm 1) is described as follows:

Algorithm 1 Ascii valueGen
Input: *Message M*
Output: *Ascii value Vi*
1: *compute length of the text Lt*
2: *compute ascii value Vi for text*
3: *record Vi at data owner*

After the message M has been received from data owner, computes its length and associated ASCII value. Since, an opponent can observing M but not having to access secret key SK or M, converting message bytes into ASCII codes is unintelligible. Determining prime/non-prime numbers (Algorithm 2) is described as follows:

Algorithm 2 Verify Prime/Non Prime
Input: *Ascii Value Vi*
Output: *Prime/Non Prime*
1: **for** *each value Vi* **do**
2: *compute prime numbers Pn*
3: **end for**
4: **for** *each value Vi* **do**
5: *compute non prime Np*
6: **end for**
7: *record Pn and Np*

When the ASCII values of the message have been calculated, prime number Pn, non-prime numbers Np are computed and recorded. This process is feasible to subdivide message into prime and non-prime group. If the opponent is interrupted for intercepting only this particular message, then he/she may be difficult to concentrate on grouping of prime/non-prime. And the matrix creation algorithm (Algorithm 3) is described as follows:

Algorithm 3 Matrix Creation
Input: *Pn, Np*
Output: *matrix formed by SK*
1: *compute row and column values*
2: *compute matrix* Mat_i
3: *generate secret key SK*
4: **for** *each* Mat_i **do**
5: *compute* $\{ Mat_i \ || \ SK\}$
6: **end for**
7: *record final matrix* Mat_j

An algorithm 3 forms a square matrix Mat_i with the combination of prime/non-prime numbers. The matrix Mat_j is obtained by concatenating between Mat_i and one time secret key SK. Based on the double transposition cipher, matrix containing cipher text values have been generated and recorded. The CiphertextGen algorithm (Algorithm 4) is described as follows:

Algorithm 4 Cipher textGen
Input: Mat_j
Output: *complete cipher text C*
1: **for** *each* Mat_j **do**
2: *compute string character* V_j
3: **end for**
4: *record* $C \leftarrow V_j$
5: *record C at public or private cloud*

Once the ciphertext matrix has been computed, we translated all ciphertext values into sequence of string characters Vj and that have been represented as final ciphertext C. From algorithm 1-4, we just computed an encrypted message; later in the stage that message is uploaded or stored on CSP or public or private cloud. Decryption for generating original data is a reverse process of encryption.

IV. EXPERIMENTAL RESULTS

We have compared the performance of existing algorithms and proposed method using Cloud Simulator 3.0.3 and dell i3 processor of 4GB RAM. The network parameters which are taken in to considerations are shown in following Tables. Table II depicts the time taken in terms of millisecond of Advanced Encryption Standard (AES) along with message bytes. As key length of AES is 128 bits and number of rounds to generate ciphertext is relatively increased, the time taken to compute resulting ciphertext is also increased. The graphical representation of Table II is depicted in Fig. 2. Here, we tested AES algorithm on cloud based and network based examples of cloud simulator.

Table II: AES Time Taken (ms)

Size of file (Bytes)	34	78	154	189	215
CE-1	524	514	534	556	518
CE-2	520	513	517	550	518
CE-3	509	506	517	542	511
CE-4	512	525	531	524	520
CE-5	502	523	510	543	519
CE-6	552	551	546	547	537
CE-7	517	515	512	533	523
NE-1	545	520	512	540	521
NE-2	518	533	519	559	507
NE-3	513	531	506	519	543
NE-4	517	517	505	532	515

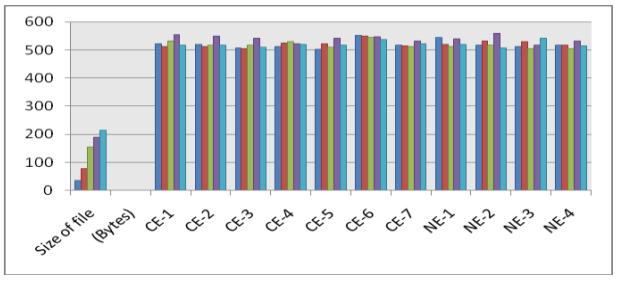

Fig. 2: AES Computational Efforts

© Copyright: All rights reserved.

AES uses conventional locking system where both sender and receiver need an identical copy of key to the encryption and decryption. Thus, sender and receiver need to share the same key. This is analogous to AES cryptosystem. Table III depicts the DES time taken in terms of milliseconds. As key length, number of rounds of DES is practically less compared to AES, the computational efforts is shown in Fig. 3 is differs.

Table III: DES Time Taken (ms)

Size of file (Bytes)	34	78	154	189	215
CE-1	411	407	415	418	395
CE-2	400	397	400	404	416
CE-3	418	406	405	417	419
CE-4	408	418	405	428	412
CE-5	408	408	407	401	412
CE-6	466	492	518	571	914
CE-7	445	502	521	566	584
NE-1	414	422	396	416	417
NE-2	411	401	423	414	411
NE-3	421	422	407	406	433
NE-4	428	396	417	410	400

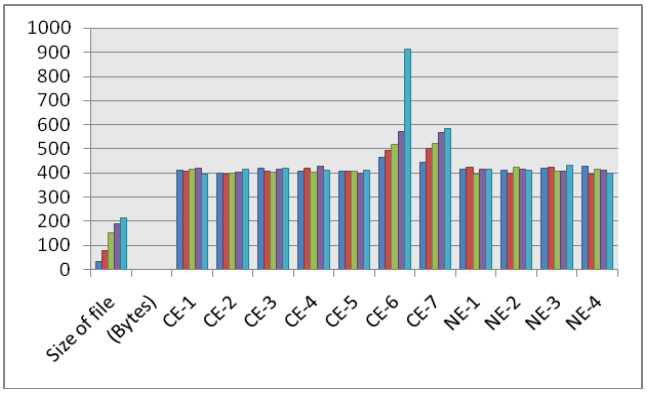

Fig. 3: DES Computational Efforts.

The ability to make encryption algorithm makes the concept of cryptography seem extremely attractive for a number of different applications and situations. As we seen the computational efforts of AES and DES algorithms is extremely high in number, our proposed method is practically reduces the cost of computational efforts is depicted in Table IV and its corresponding graphical representation shown in Fig. 4. From the experimental results we can adopt our method in real world applications with affordable cost. (CE- indicates cloud example and NE- indicates network example in Cloud Simulator 3.0.3).

Table IV: Proposed Method Time Taken (ms)

Size of file (Bytes)	34	78	154	189	215
CE-1	15	25	23	27	91
CE-2	31	24	27	32	91
CE-3	15	18	26	26	91
CE-4	15	40	27	26	91
CE-5	16	33	25	26	55
CE-6	31	40	43	59	83
CE-7	47	26	95	32	121
NE-1	31	31	25	27	68
NE-2	15	53	24	40	116
NE-3	16	21	26	25	70
NE-4	16	71	24	22	123

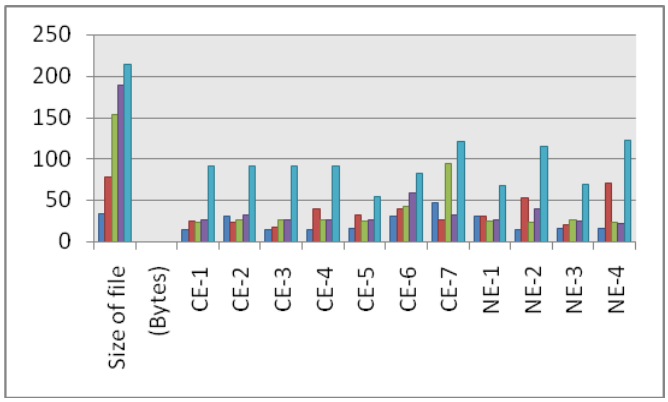

Fig. 4: Proposed Method Computational Efforts.

A. Computational Analysis (wrt DES NE-2)

As part of computational analysis, now we validate our proposed method with simulated values. In a modern processing system, if we reduce the computation time for encryption and decryption that leads to increase the throughput intern performance of the system are also increases. By this we provide the service to the intended user within stipulated time and we avoid the possible real time attacks. We computed following equations for the analysis purpose:

$$\text{Percentage Increase (PI)} = T * 100 \quad (1)$$

$$T = \frac{V_{old} - V_{new}}{V_{old}} \quad (2)$$

Table V illustrates the comparison of proposed method along with DES algorithm using equations (1) and (2). The proposed method has been validated approximately 88 percent of higher execution speed compared with simulated values of DES.

Table V: Analysis of proposed method vs DES

Vold	Vnew	T	PT
411	15	0.9635	96.35
401	53	0.8678	86.78
423	24	0.9432	94.32
414	40	0.9033	90.33
411	29	0.9294	92.94

Retrieval Number: J90300881019/19©BEIESP
DOI: 10.35940/ijitee.J9030.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

B. Computational Analysis (wrt AES NE-2)

Table VI depicts the computational analysis of proposed method along with AES algorithm using equations (1) and (2). There four, proposed method has been validated approximately 90 percent of higher execution speed compared with simulated values of AES.

Table V: Analysis of proposed method vs AES

Vold	Vnew	T	PT
518	15	0.9710	97.10
533	53	0.9005	90.05
519	24	0.9537	95.37
559	40	0.9284	92.84
507	29	0.9428	94.28

C. Computational Analysis (wrt DES NE-2)

In cloud service based environment, complexity limits the quality of service requirements such as average response time, reliability, numerousness and uncertainty, these measures the actual running performance of cloud applications. Entropy based methodology is appropriate to avoid and reduce the complexity [17]. In this paper, we measured entropy values of proposed method and existing algorithms using equation (3). Table VII shows the entropy comparison, which shows that entropy value of proposed method is comparatively reduced while comparing with DES, AES in order to avoid and reduce the complexity.

$$E(D) = \sum_{k=0}^{n} p(D_i) \log_b p(D_i) \quad (3)$$

Table VII: Entropy Evaluation

Size of file (Bytes)	DES	AES	Proposed Method
34	5.12	3.82	4.07
78	5.91	3.92	3.72
154	6.40	3.94	2.52
189	6.74	3.98	4.28
215	6.82	3.96	1.0

V. SECURITY MEASURES

A. Throughput

As huge data has to be processed on the cloud environment, calculating the throughput of encryption and also decryption is reflected on the power consumption [10]. In this paper, we computed throughput using equation (4), if this value is increased, then the power utilization of the algorithm is decreased. Table VIII shows the throughput evaluation, in which throughput value of proposed method is increased goes on while comparing with AES, DES algorithms shown in Fig. 5.

$$\text{Throughput} = \frac{\text{Number of Bytes Encrypted}}{\text{Time Taken}} \quad (4)$$

Table VIII: Throughput

Size of file (Bytes)	DES	AES	Proposed Method
34	0.08	0.06	2.26
78	0.19	0.19	4.33
154	0.37	0.28	5.92
189	0.45	0.33	7.26
215	0.54	0.41	7.67

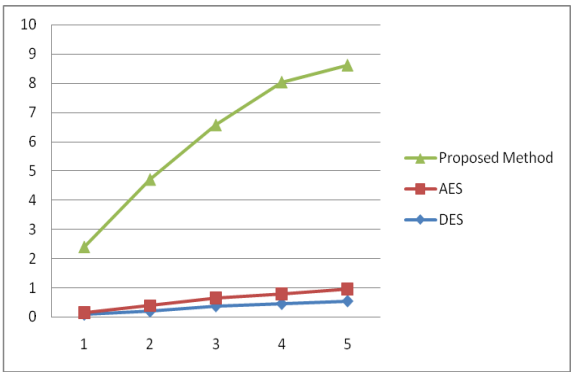

Fig. 5: Throughput Evaluation

B. Avalanche Effect

Avalanche effect describes the significant changes is ciphertext whenever the changes occurs in plain text. It is measured using hamming distance and it helps to determine the dissimilarity between the plain text and ciphertext changes [10]. In this paper, we measured the avalanche effect by determining hamming distances among AES, DES and proposed method shown in Table IX. Fig. 6 shows the hamming distance, it reflects the performance of proposed method is almost identical with respect to DES, AES algorithms.

Table IX: Hamming Distance

Size of file (Bytes)	DES	AES	Proposed Method
34	33	34	30
78	77	60	75
154	152	154	154
189	184	187	186
215	212	208	213

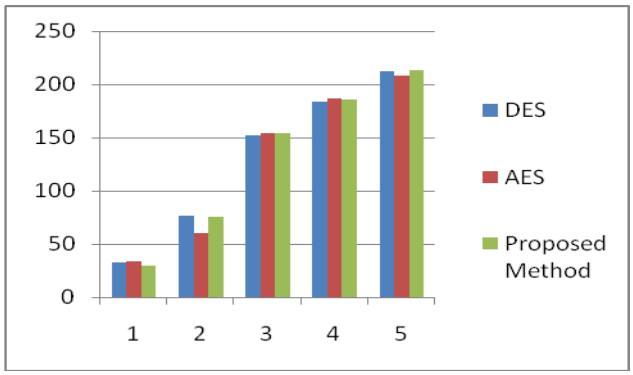

Fig. 6: Hamming Distance

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

VI. CONCLUSION AND FUTURE WORK

The proposed method provides a possible solution to data leakage for which better confidentiality is increased. An experimental result shows that the proposed method is efficient and effective. We have shown that proposed method requires less processing time, less entropy and it relatively reduced processing overhead to run this system on cloud platform. We have estimated the throughput of proposed method; it is practically increased in comparison with existing algorithms. To increase the confidentiality, we also measured the avalanche effect that satisfies the targets of existing algorithms. Thus, the ciphertext generated from the proposed scheme may be unpredictable and output not depends on any specific key. So the prediction or cracking of data by the intruder may be intangible that is the major strength of our technique. This method is experimented on cloud simulator, in future, we will experiment on a real cloud environment and also IOT based applications.

ACKNOWLEDGMENT

We would like to express our special thanks to Research Centre, Department of Computer Science and Engineering, Principal, J N N College of Engineering, Shimoga for their encouragement to take up our research work. We would also like to thank HOD, Principal, G M Institute of Technology, Davanagere for their valuable suggestions to take up our research work.

REFERENCES

1. R. Sugumar and S. B. S. Imam, "Symmetric encryption algorithm to secure outsourced data in public cloud storage," *Indian Journal of Science and Technology*, vol. 8, no. 23, 2015.
2. M. F. Mushtaq, S. Jamel, A. H. Disina, Z. A. Pindar, N. S. Ahmad, and M. M. D. Shakir, "A survey on the cryptographic encryption algorithms," Proceeding of (IJACSA) *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 11, 2017.
3. S. Aldossary and W. Allen, "Data security, privacy, availability and integrity in cloud computing: issues and current solutions," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 4, pp. 485–498, 2016.
4. B. Feng, X. Ma, C. Guo, H. Shi, Z. Fu, and T. Qiu, "An efficient protocol with bidirectional verification for storage security in cloud computing," *IEEE Access*, vol. 4, pp. 7899–7911, 2016.
5. A. Sachdev and M. Bhansali, "Enhancing cloud computing security using aes algorithm," *International Journal of Computer Applications*, vol. 67, no. 9, 2013.
6. C. Kaur and E. G. S. Bhathal, "Data security algorithms in cloud computing: A."
7. Z. Xiao and Y. Xiao, "Security and privacy in cloud computing," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 843–859, 2013.
8. M. G. Durga et al., "Study on data security mechanism in cloud computing," *in Current Trends in Engineering and Technology (ICCTET), 2014 2nd International Conference on. IEEE*, 2014, pp. 13–17.
9. M. Ali, R. Dhamotharan, E. Khan, S. U. Khan, A. V. Vasilakos, K. Li, and A. Y. Zomaya, "Sedasc: secure data sharing in clouds," *IEEE Systems Journal*, vol. 11, no. 2, pp. 395–404, 2017.
10. P. Patil, P. Narayankar, D. Narayan, and S. M. Meena, "A comprehensive evaluation of cryptographic algorithms: Des, 3des, aes, rsa and blowfish," *Procedia Computer Science*, vol. 78, pp. 617–624, 2016.
11. F. Maqsood, M. M. Ali, M. Ahmed, and M. A. Shah, "Cryptography: A comparative analysis for modern techniques," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 6, pp. 442–448, 2017.
12. M. Ebrahim, S. Khan, and U. B. Khalid, "Symmetric algorithm survey: a comparative analysis," *arXiv preprint arXiv*:1405.0398, 2014.
13. M. F. Mushtaq, S. Jamel, K. M. Mohamad, S. K. A. Khalid, and M. M. Deris, "Key generation technique based on triangular coordinate extraction for hybrid cubes," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 9, no. 3-4, pp. 195–200, 2017.
14. M. R. Baharon, Q. Shi, M. F. Abdollah, S. W. M. S. Yassin, and A. Idris, "An improved fully homomorphic encryption scheme for cloud computing," *International Journal of Communication Networks and Information Security*, vol. 10, no. 3, p. 502, 2018.
15. K.-K. R. Choo, O. F. Rana, and M. Rajarajan, "Cloud security engineering: Theory, practice and future research," *IEEE Transactions on Cloud Computing*, no. 3, pp. 372–374, 2017.
16. S. Singh and A. Sharma, "Analysis of endecloudreports for encrypting and decrypting data in cloud," *International Journal of Computer Applications*, vol. 136, no. 12, 2016.
17. H. Chen, F. Z. Wang, and N. Helian, "Entropy4cloud: Using entropy-based complexity to optimize cloud service resource management," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 13–24, 2018.

AUTHORS PROFILE

Shivanna K received the B.E. degree in Computer Science and Engineering from Visvesvaraya Technological University, Belagavi, in 2006, and M.Tech. degree in Computer Science and Engineering from the Visvesvaraya Technological University, Belagavi, in 2010. He is currently an Assistant Professor with Computer Science and Engineering, GMIT, Davanagere. His current research interests include cryptography, network security, and in particular, cloud computing security.

Dr. Prabhudeva S received the B.E. degree in computer science and engineering from University of Mysore in 1990, the M.S. degree in system and information from the Birla Institute of Technology and Science, Rajasthan, in 1996, and Ph.D. from the IIT, Bombay, in 2011. He is currently working as a professor in the Department of Information Science and Engineering, JNNCE, Shimoga. He has published more than 15 papers. His research interests include network security, software reliability, and dependability modeling. He has served as an HOD of the Department of Information Science and Engineering, and also served as a Professor in Computer Science and Engineering, JNNCE, Shimoga. He is currently member of Board of Studies, and also a LIC committee member in Visvesvaraya Technological University, Karnataka, India. He organized several seminars and workshops in JNNCE, Shimoga.

© Copyright: All rights reserved.