International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019

Secure and Smart Monitoring of Sensitive Data in Cloud Computing using Effective Cryptographic Scheme

A.S. Kalyana Kumar, T.Abdul Razak

Abstract:*The advantage of storing infinite data without giving any attentiontothe available storage restrictions and the liberty that one possesses to use those data when needed from anywhere in the globe, makes cloud environment the most ideal technology. Cloud hadalso became the preferred platform for storing and transferring the intendeddata. Both individuals and organizations were much comfortable in letting their sensitive data across their own generated cloud environments. But apart from the advantages that the cloud environments have imparted, it also have its own limitations. Limitations like risk of maintaining user's data privacy, integrity, confidentiality and safety were normally existing while using the cloud environments. Besides allof the above said challenges, transferring of data from organization to cloud servers was considered with major importancein many evolving works. Several algorithms and encryption techniques have been developed currently for securing the operation of data transmission. Along this way, this paper proposes a design for thecloud framework that guarantees secured data transmission to serverfrom the source. This paper utilizes an approach called Honey Bee algorithm for encrypting and decrypting the datato be transmitted to the server. This algorithm is basically an encryption scheme which provides flexibility againstthe potential attacks by delivering a reasonable-looking text. But, the reasonable-looking text is a kind ofplain text generated for every wrong key used by an attacker for decrypting the message. Here a count of five keys will be generatedby the* **PRNG algorithm***right after matching the OTP keys, which will be thenrandomly sent to users. The generated five keys aredeployed to promote additional security by the operation of anonymizationagainst the intruderstrying to access the data while under transmission. Here, in this proposed methodology,user cannot view all of his/her information, whereas the admin can have the full privilege of viewing all information about a user.Furthermore, performance metrics such as Key generation time, Encryption time, Decryption time and Message time were taken for the performance analysis to validate the performance of this proposed methodology with the existing considered methodologies.*

Keywords:*Cryptography, Honey Bee Algorithm, Encryption, Decryption, Key Generation, access control, privacy preserving.*

Revised Manuscript Received on October 30, 2019.

* Correspondence Author

A.S. Kalyana Kumar *, Research Scholar, Research and Development Centre, Bharathiar University, Coimbatore Tamil Nadu, India. Email: kalyanakumarphd8@gmail.com

DR. T. Abdul Razak, Professor, Department of computer science and engineering, Jamal Mohamed College, Tiruchirapalli, Tamil Nadu, India.

I. INTRODUCTION

Cloud environment [1] delivers various supports to resolve the difficulties that we are facing with shared computing resources namely networking, software analysis, etc. These applications has developed into Big data analysis, IoT, and so on. It has several advantages namely users can manipulate, configure, and access the data over the internet any time. Moreover, it does not require any sort of software installation to access the cloud. Due to the vast range of applications, people are now paying more attention towards cloud security for managing, storing, and processing data. Besides innovation, cloud environment faces issues concerned upon security. The data storage in cloud is provided by third party service providers. Thus, sometimes it is a risky entity to store sensitive data to such service providers. Cryptography [2] plays an important role for fulfilling the above mentioned demands. It involves encryption and decryption of data. For the past few decades,various researchers have developed a large number of encryption and decryption techniques such as RSA, AES, etc. But these conventional researches fails to resolve some limitations like maintaining robust security between communication channels, Packet delay, and so on. Managing key generation is one of the major aspects in cryptographic mechanism [3]. It is the process of secured administration of private, public, master and secret keys. Generally, a key is a string of binary 0s and 1s and it is used in crypto algorithms for decryption and encryption. Threats will be raised, if the keys are improperly managed. Recently, Honey Encryption (HE) is deployed widely. This scheme is used for enhancing the security of encrypted data [4].

In HE, when an intruder tries to access the encrypted data using incorrect password, instead of rising access errors, the HE algorithm generate a bogus data to look plausible. Such that, the intruder will be puzzled in determining whether the system is accessed or not. In order to preserve the privacy of most sensitive data, the user will encrypt it before sending them to cloud server [5]. After that, the cloud will do some alterations over the resulted encrypted data like feature extraction, face recognition, etc. This privacy preserving scheme will provide anonymity to the sensitive data and hide the essentials even to the appropriate receivers.

There are some problems that are identified in existing methodologies:

- Robust security
- Key length and plain text length should be maintained same
- Increased execution time
- Increased computational cost
- Insufficient key generation

In this proposed work, Honey Encryption and Decryption is utilized for encrypting and decrypting the information. While encrypting, keys will be generated (five keys) using Pseudo Random Number Generator (PRNG) for providing additional security. If an intruder tries to access the data using one of the keys, then another key will be verified by rejecting the first key. Finally, anonymity selection is utilized for hiding the sensitive data even after encryption. Thus sensitive information will get secured.

The primary objectives are as follows,

- To maintain potential security against intruders by using Honey Encryption and decryption
- To maintain parallelism between the key and plain text length
- To reduce execution time and computational cost
- To generate number of keys (5) for additional security.

The remaining portion of this paper is summarized as follows: Section II demonstrates the conventional works associated to the cryptographic technique using various methodologies. Section III illuminates about the proposed Honey Encryption and decryption methodology. Section IV exhibits the performance study of the projected mechanism and in conclusion, Section V concludes the proposed work.

II. RELATED WORKS

Cryptography is an art of converting plain data into a secured data so as to eliminate attacks by unauthorized user. The original data before conversion is said to be cipher text. The term encryption is used for transforming the plain data or plain text into cipher format and decryption is used for transforming cipher text into plain text. It is well known that, encryption algorithm is used in sender side and decryption in receiver side. There are three types of cryptographic algorithms namely symmetric key, Asymmetric key and Hashing cryptography algorithms[6].

A. Symmetric key Cryptography

This type of cryptography utilizes two distinct algorithms for encrypting and decrypting the data. A key is generated and this same key is used by both sender and receiver for encryption and decryption of data respectively[7]. Some of the existed works for representing this symmetric key are provided below.

[8] designed a secured cryptographic AES accelerators using information flow such that the security is assured. Advanced Encryption Standard (AES), a symmetric cipher standard widely used for ciphering the sensitive data. By using 128/192/256 bit key, AES will encrypt 128-bit plain text into 128-bit cipher text. In this related work, AES accelerator prototype is used such that both efficiency and security is attained together. It used runtime policies for providing flexibility for practical usage and design time policies ensured guarantee for security.

[9]proposed Data Encryption Standard (DES) and 3DES methods in Near Field Communication (NFC). Commonly used cryptographic methods for preventing text messages were DES and 3DES. This paper also explained about the mechanism and its security role in smart cards. Various things like working methodology of DES and 3DES in data protection by creating applications which utilized CPP language in encrypting data writing process to smart cards and decrypting data reading process from smart cards. This paper concluded that the DES' run time of writing and reading process using smart cards is faster than the 3DES methodology.

[10] implemented 3DES framework for various block cipher algorithms for symmetric cryptography. This algorithm define the steps mathematically which was required for transforming data into cipher text and also to transform vice versa. This proposed framework is designed using VHDL language and the results are generated using a software tool called Xilinx. The secured text (both encrypt/decrypt) was 192 bits. But without keys, it would be 168 bits. It had been observed that, 3DES is completely safe and exponentially potential than DES.

[11] proposed a novel methodology in securing cloud by using Blowfish algorithm and MD5. This paper stated that, there was no variation of encryption in cloud or client aspects. This might resulted in minimized security. Thus, this paper came up with a novel cryptographic approach by molding MD5 and Blowfish schemes along with hybrid based functionalities, which in turn upgraded the security. A calculation is constructed to eliminate the limitations from symmetric key cryptography and hash function schemes.

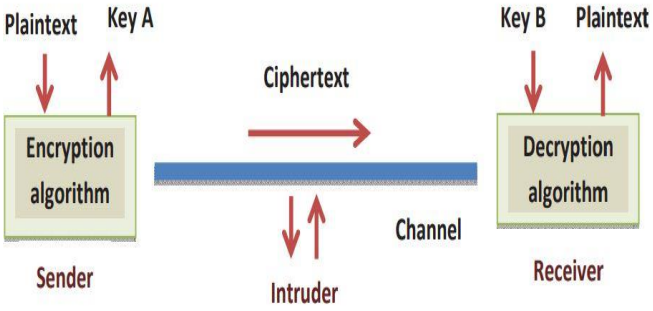

Fig. 1. Procedure followed in Encryption and Decryption

[12] developed an enhanced RC4 algorithm to perform encryption parallely using Hadoop. Generally, RC4 is an effective cipher algorithm used for encryption. A vast requirement was introduced in securing big data. Usually, Hadoop had been a weaker method to ensure the privacy and security of the data. Thus this paper modified the RC4 scheme for enhancing the strength of Hadoop security. MapReduce is used for minimizing the cost of encryption and finally a key is generated.

B. Asymmetric key Cryptography

This type also makes use of two distinct algorithms for encryption and decryption. Unlike symmetric, two keys namely public and private keys for encryption and decryption respectively are generated. Existing works for representing asymmetric key cryptography are provided below,

[13] developed an efficient memory scheme by modifying the RSA methodology in order to transport sensitive data across IoT and Cloud securely. As it is an asymmetric type, it involves two keys namely private and public keys. To enhance the memory efficiency, this work reused the RSA method along with Diophantine of nonlinear formulation. Due to the addition of RSA into the research work, the performance level is increased. Moreover, this represented work did not require any multiplicative inverse function or any other Euclid's algorithm. The experimentation had been done on various phases of MEMK PKC like encryption, decryption, and key generation by altering N-bit modulo bits from 1000 to 10000. [14] represented a novel quantized key distribution and cryptographic mechanisms for securing cloud data. This paper stated that securing health records in cloud is critical. Various security mechanisms were developed for the purpose of focusing authentication based security. This paper developed a Non-Abelian Encryption (QKD-NAE) to secure and to access PHR. Here, the admin and the user utilized the quantum key to access data. The admin of the hospital used Diffie Hellman (DH) for the purpose of securing key generation and every user separate the key based on respective designation. By using NAE, the admin divided the input into attributes for encryption. Here the access control is taken care by Third Party Auditor (TPA). The appropriate user received the data from various cloud servers based on access provided by DH algorithm. The main advantage if this work is that intruders were not able to know about the generated quantum key and the generated message.

[15] constructed a homomorphic encryption for outsourced images in mobile and cloud computing. As the cloud and mobile got integrated and resulted in mobile cloud computing (MCC), many benefits were offered like capacity, scalability, reliability, and so on. The image data processing and data storage were migrated into cloud environment. Thus to secure the outsourced images, an architecture is proposed where two clouds were existed. Private cloud which was dedicated for encryption and decryption and public cloud dedicated for storage. The private cloud was developed using openstack. The encryption scheme is designed using Paillier's cryptosystem - a designated design for images. The property of homomorphism is evaluated by using DWT algorithm.

C. Hashing Cryptography

Hash functions are considered to be a base in the field of cryptography. These functions are widely utilized in important applications in the field of authentication, integrity, digital signatures, time stamping, etc. The function is denoted as H. It will be computed by taking inputs as arbitrary length message (M) and fixed length key (K) and provided output called message digest (D). Some of the widely used hashing functions are provided below,

[16]designed a Dynamic Proof of Storage (PoS) for multi user cloud environment by using MD5 (Message Digest 5). PoS is a beneficial cryptographic technology that enhanced the coherence between the client and respectable outsourced servers. A cross-client replica method is needed by customer side to avoid transferring the procedure, when the similar documents had been transferred to server. This research work provided an adaptable and helpful path for sharing information, which in turn brought various advantages for both public and private. This would resulted in characteristic protection for both outsource and insource of data. Thus a character based encryption is assembled for handling information sharing. [17]developed a cryptographic mechanism for storing data in cloud. It utilized MD6 along with unscrambling and encryption. Cloud figuring framework consisted of virtualized and interconnected frameworks. Several investigations have been conducted on cloud security which paved the way that were to be migrated into the cloud. Thus this paper came up with a security mechanism that improved key generation algorithm. Various experimentation had been carried out that shown the proposed technique provided better results than other conventional approaches.

[18] dealt various impacts like 3-D Radiated effects, Single Habit assumption (SHA), and inhomogeneity's in Cloud environment by using Moderate Resolution Imaging Spectrometer (MODIS). It collected six cloud products with errors resulted due to SHA, Plane Parallel Assumption (PPA), etc. The term SHA denoted that each and every Cirrus cloud was assumed to have similar shape of ice crystals. Thus a method is proposed to single out these six type of errors. These type of errors were evaluated using Spherical Harmonics Discrete Ordinate Method for MODIS. On the average level, the errors occurred in SHA were within 16%–43% (39%–2%), but the 3-D RE- and cloud inhomogeneity- errors in COT were within 6%–20% (3%–8%) and 2.7%–0% (4%–10%), respectively.

[19] constructed a security framework based on blockchain for distributed cloud storage. Here, the users divided their respective files into encrypted data parts, and then these parts were uploaded in a random manner into the P2P network which provided free storage capacity. In SHA, the performance depended upon the length of hashed text. SHA 256 utilized bit coins for generating 256 output from input. This operation had a pseudo-randomness and its calculation is very irregular such that input is altered and the resulting output will vary in wider range. Every head of the blocks stored a Merkle 295 root for verifying the files. Thus the entire transactions can be verified by this blockchain.

The comparative summary of above methods are provided in [20]. These algorithms are majorly depend upon factors like block size, key length, rounds, contributor, etc. From the provided survey, it is seen that the existing techniques and algorithms have both advantages and demerits, but it mainly lacks with the following limitations like:

- Key length and plain text length should be maintained same
- Increased execution time
- Increased computational cost

To resolve these limitations, this work aims to propose a cryptographic strategy in cloud.

III. PROPOSED WORK

The overall novelty of the proposed work is shown in Fig. 2. It has two segments namely user and admin.

The admin has the full privilege for viewing and updating the user information. The user should send request to admin for viewing or updating his/her information. After these operations, the information will be encrypted by using Honey encryption method (HE).This encrypted data will be sent to user. An OTP matching is also taken up before generating the keys for the sake of matching the intended OTPs.Meanwhile, keys (5 keys) will be generated using PRNG (Pseudo Random Number Generator) which in turn uses mathematical formulae for generating random approximated sequential numbers. Out of five keys, single key will be send for verification. Once it is matched, the user information will be permitted for decryption. If any intruder sends the key, then another key (from the remaining four keys) will be verified and eliminates the usage of previous key by making it useless. Thus making the system more secure against attackers, by preventing them on viewing the information.

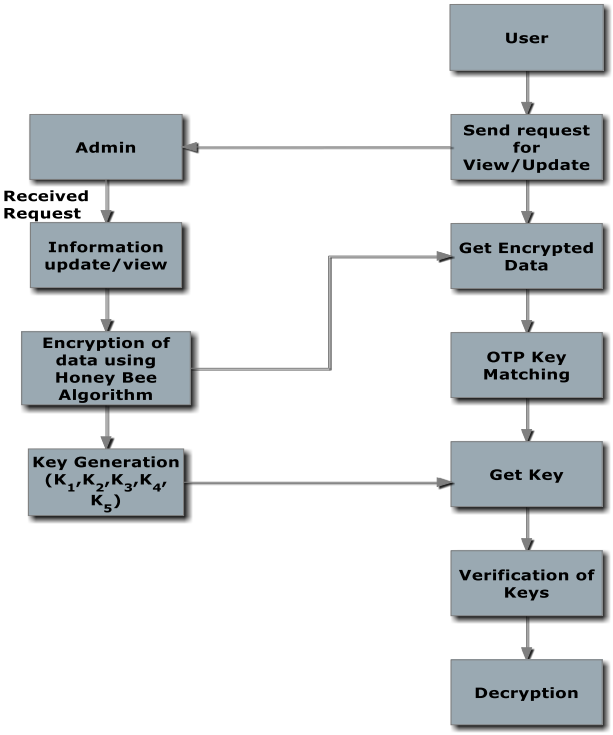

Fig. 2. Overviewof proposed work

A. Enhanced Honey Encryption Scheme

This HE methodology has syntax and semantics similar to that of symmetric encryption scheme. Encryption is used for mapping the generated key and corresponding message/information to a ciphertext. In this research work, the encryption will be in random manner. Before encryption, data preprocessing should be done in order to eliminate the noise and other disturbances. The pre-processed data, then undergoes encryption. Consider $\mathcal{K}$and$\mathcal{M}$ as key and message spaces respectively. In general, it is assumed that $\mathcal{K}$ consists of bit strings with variable length. A HE scheme covers both encryption and decryption of information. This pair of algorithms can be denoted as $HE = (HEnc, HDec)$. The encryption will input a key $\mathcal{K} \epsilon \kappa$, and message as $\mathcal{M}\epsilon m$, some of the randomized uniform bits, and finally outputs the ciphertext C. This ciphertext can be represented as $C \leftarrow \$HEnc\mathcal{K}(\mathcal{M})$, where $\leftarrow \$$ represents that $HEnc$ might use number of randomized uniform bits. The $HEnc$ can be formulated as below,

$$\sum_{c\epsilon c} pr[M = M^*|C^*C|].pr[C^* = C] \qquad (1)$$

Where M is the message and M^* can be denoted as below,

$$M^* \leftarrow_{p_m} M \qquad (2)$$

Where p_m is the message distributedon M.

Where C is the ciphertext and C^* will be represented as below,

$$C^* \leftarrow \$HEnc(K^*, M^*) \qquad (3)$$

Where K^* represents the key distribution over K.

The enhanced HE scheme is explained in pseudo code 1.

Pseudo code 1 : Enhanced Honey Encryption
Input: Pre-process data (PRE_{dt})
Output: HONEY ENCRYPTION data (Des_{EN})
Procedure:

Step 1: text as txt
Step 2: M= PRE_{dt}
Step 3: s← encode (M)
Step 4: r ← {0,1}
Step 5: ss← (r,k)
Step 6: s← c XOR ss
Step 7: return (r,c)
Step 8: End

B. Key Generation

In this paper, PRNG algorithm is used for generating keys and delivers secure transmission of data and key. This algorithm uses mathematical formula for producing sequence of random numbers. Here the input will be Seed Key and the output will be the secret keys (k1, k2, k3, k4, k5). The seed key is fed into key generation process, initiallybefore which the OTP key matching operation takes place wherein the generated OTP will be matched. Here this seeded key I the initial value used for generating the pseudorandom integer values. By using this process, the random keys are produced while encrypting the data files. In proposed work, we are generating k1, k2, k3, k4, k5 as secret keys.

Fig. 3. Overview of PRNG in key generation.

The key generation scheme is explained in pseudo code 2.

Pseudo code 2: Key Generation
Input: *User Input*
Output: Key generation
Procedure:
Step 1: N ← sequence of random values

Step 2: XOR←exclusive-or operator
Step 3: VV ← Result key
Step 4: C ←true seed
R ← concatenated to produce a pseudo random number
Step 5: N_{x+1} = (aN_x + c) mod c

Step 6: For I in 5
m, 0< m- modulus
a, 0< a <m- multiplier
c, 0< = c<m- increment
N_0, 0<=x_0<m- the seed or start value
Step 7: I = ede*X(N_0)
Step 8: R = ede*Ktt(II XOR VV) and a new V is generated by
Step 9: VV = edt*KK (RR XOR II).

Step 10:End loop

C. Access Control Policy

Usually, the data access is controlled based on authority. There is only one user (admin) can access the data stored at cloud based on authority level.

The pseudo code of access control is explained in pseudo code 3. In our proposed work, the sensitive (S) and non-sensitive (NS) information can be seen by admin and user has the privilege of seeing non-sensitive information alone.

Pseudo code 3: Access control policy
Input: Pre-process data (PRE_{dt})
Output: Aceess Control Policy(AC_{po})
A → Admin
U → User
UI → User Information
S → Sensitive
NS → Non Sensitive

$$\boldsymbol{Des_{en}} = \begin{pmatrix} \boldsymbol{UI(7c, 2S, 5} \\ \boldsymbol{NS)} \end{pmatrix};$$

$$\boldsymbol{AU} = \begin{cases} \boldsymbol{if\ A(S, NS)} \\ \boldsymbol{else\ (NS)} \end{cases}$$

D. Enhanced Honey Decryption

The general purpose of decryption is to recover the information from cipher texts. The traditional symmetric decryption schemes involved in raising errors, when wrong key is inputted. But in proposed, if any intrusion happens, instead of rising error, the decryption simply show the plain text that looked plausible. Here the input of $HDec$ is key κ, cipher text as C and the output will be the final information of user $\mathcal{M}\epsilon m$. Decryption can be described as below,

$$\mathcal{M} \leftarrow HDec\mathcal{K}(C) \quad (4)$$

Honey based decryption can be evaluated from below formulae,

$$\sum_{c\epsilon c} pr[M = decode(K^*, C)|C^* = C].\frac{1}{|C|} \quad (5)$$

$$\sum_{c\epsilon c} L_{HE.PK}(C).\frac{1}{|C|} = E[HE_{PK}] \quad (6)$$

The Honey based decryption will be worked as, $[M = decode(K^*, C)|C^* = C].\frac{1}{|C|}$, for all$K\ \epsilon\ \kappa$, $M\epsilon m$ and the values used by $HEnc$.

The enhanced Honey Decryption scheme is explained in pseudo code 4.

Pseudo Code 4: Enhanced Honey Decryption
Input: *ENCRYPTION data (Des_{EN})*
Output: *Decryption (ENC_{Ds})*
Step 1: text as txt
Step 2: $txt = PRE_{dt}$
Step 3: HDec (k,(r,c)
Step 4: ss← c XOR ss
Step 5: M ← decode (s)
Step 6: return M
Step 7: End

E. Anonymization using Selection-anonymity

In this privacy preserving, the major requirement of security is to hide the access related ciphertext against the appropriate receivers. None of the sensitive information should be derived from the ciphertext. The process of Anonymization is represented in pseudo code 5.

Pseudo code 5: *Anonymization using Selection-anonymity*
Input: *set of Participant Attributes*
Output: *Anonymized Attributes*
I ← age
N ← no of information
O ← Occupation
J ← Anonymized age values
R ← random number generation
n ← file size
Step 1: generator random number
Step 2 : for I to n
Step 3: x ← random number generator
Step 4: l ← x
Step 5: o ← x
Step 6: j ← x
Step 7: o ← x
End loop

IV. PERFORMANCE ANALYSIS

In this section, the experimental results of existing and proposed techniques are evaluated and compared by using various performance measures, which includes encryption time, decryption time, key generation. Also, the existing techniques such as RSA, DH, and DSA are considered to prove the effectiveness of the proposed Honey encryption and decryption technique.

A. Key Generation Time

It is defined as the quantity of time that proposed work taken for data transferring and its execution. It is evaluated as follows:

$$Key\ Generation\_Time = Data\ Transferring\_Time + Execution\_time \quad (7)$$

The comparison facilitating data between the existing and proposed methodologies for the Key generation time is indicated in the below table 1.

Table 1. Comparable data for the Key Generation time in both the existing and proposed methods

Key Length	RSA	DH	DSA	Proposed
512	0.132	0.232	1.236	0.00215
1024	0.431	0.421	1.9632	0.03154
2048	2.259	1.526	2.0548	1.0542

Key Generation Timeis calculated with respect to varying number of data size and the graph is represented in Fig. 4. In this evaluation, it is proved that the proposed work requires the less key generation time by using the Honey encryption/decryption algorithm.

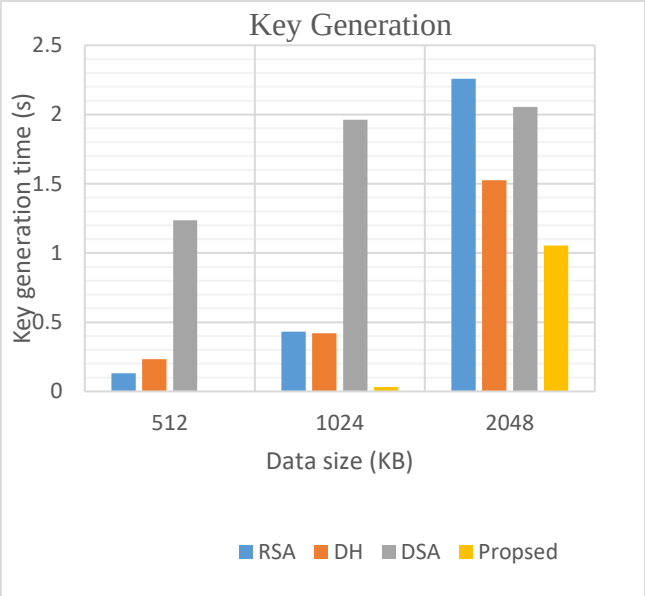

Fig. 4. Comparison of proposed method with existing methods using Key generation time.

B. Encryption Time

It is the total amount of time taken for producing a cipher-text from plain message. This time is then used for calculating the throughput of the algorithm.

The comparison enabling data between the existing and proposed methodologies for the Encryption time is shown in the below table 2.

Table 2. Comparable data for the Encryption time in both the existing and proposed methods

Message Size (bit)	Triple DES	HE AES	HE_BLowFish	Proposed System
32	2.6521	2.7892	1.8265	0.00530717
56	3.2541	3.2541	2.564	0.963254
48	4.5212	3.9632	3.1025	1.563
112	4.9632	3.0254	3.561	1.9632

Encryption Timeis calculated with respect to varying number of data size and the graph is representedin Fig. 5. In this evaluation, it is proved that the proposed work requires the less encryption time by using the Honey encryption algorithm.

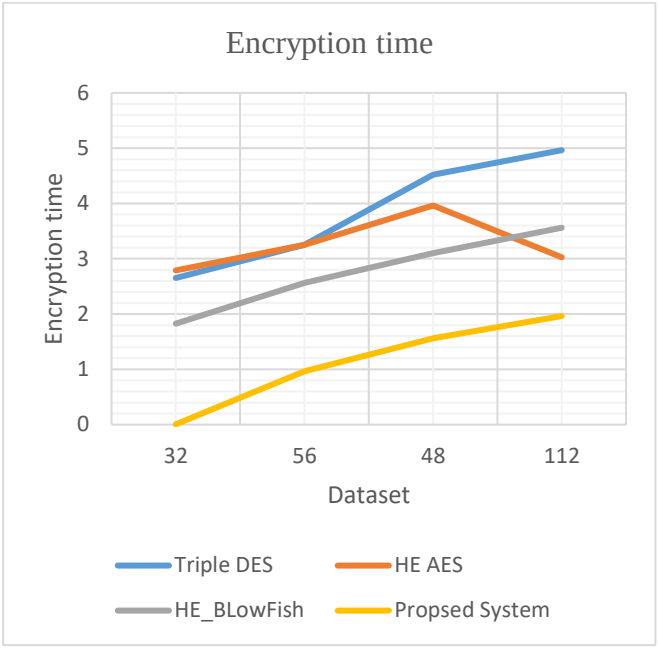

Fig. 5. Comparison of proposed method with existing methods using Encryption time.

C. Decryption Time

It is the total amount of time taken for producing the plain-message from Cipher-text. This calculated time is then used for evaluating the throughput of the decrypted algorithm.

The comparison facilitating data intermediary to the existing and proposed methodologies for the Decryption time is indicated in the below table 3.

Table 3. Comparable data for the Decryption time in both the existing and proposed methods

Message Size (bit)	Triple DES	HE AES	HE_BLowFish	Proposed System
32	1.9632	1.836	1.568	0.001670986
56	2.3698	2.1245	2.365	0.85421
48	3.256	2.1254	3.0125	2.5632
112	3.9632	3.8546	3.0254	2.8542

Decryption Timeis calculated with respect to varying number of data size and the graph is represented in Fig. 6. In this evaluation, it is proved that the proposed work requires the less decryption time by using the Honey decryption algorithm.

Fig. 6. Comparison of proposed method with existing methods using decryption time.

D. Message Time

When compared to existing technique[21], the proposed method requires the minimized execution time. It is calculated as follows:

$$Execution\_Time = Ending\_time\ of\ the\ process - Starting\_time\ of\ the\ process \quad (8)$$

The comparison enabling data betweenthe existing and proposed methodologies for the message time is indicated in the following table 4.

Table 4. Comparable data for theMessagetime in both the existing and proposed methods

Message Size	10	20	50
Ekta Agrawal (Time)	425	848	2062
Proposed (Time)	150	500	620

Also, the total execution time of the existing and proposed methods are evaluated and represented in Fig. 7. In this evaluation, it is proved that the proposed methodology requires less total execution time, when compared to the other techniques.

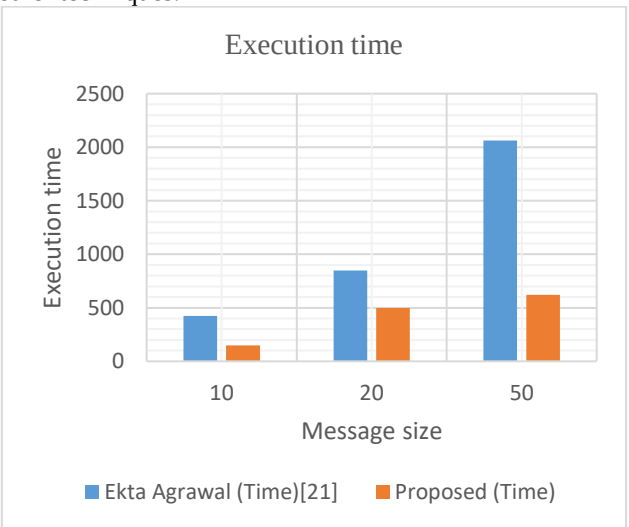

Fig. 7. Comparison of proposed method with existing methods using decryption time.

V. CONCLUSION

In this proposed work, an enhanced Honey based encryption and decryption was developed by using the key generation and selection based Anonymization. Key generation was done by using PRNG algorithm by the generation of five keys. Thus, when an intruder triedto access the user information by using one of the five keys generated, the adminwouldsent another key for verification, thusmaking the first key useless. Hence, intruder will be bewildered, whether the information was accessed or not. This utilized anonymity algorithm hided the ciphertext associated with sensitive information even against legitimate receivers/users. As a result that, none of the sensitive information will be derived from the ciphertext. This paper on the whole made use of appropriate encryption, key generation, access control policy, enhanced decryption and selection based Anonymization methodologies tosecure the data while under transmission. From the performance analysis, it could be observed that the proposed approach outperforms the other existing methodsin terms of Key generation, Encryption, Decryption and Message times. In future, these implemented methodologies will be concentrating onfurther increasing therobustness to cloud securitywith some case studies.

REFERENCES

1. P. Li, J. Li, Z. Huang, T. Li, C.-Z. Gao, S.-M. Yiu, *et al.*, "Multi-key privacy-preserving deep learning in cloud computing," *Future Generation Computer Systems*, vol. 74, pp. 76-85, 2017.
2. O. M. A. Al-Hazaimeh, "A new approach for complex encrypting and decrypting data," *International Journal of Computer Networks & Communications*, vol. 5, p. 95, 2013.
3. L. Shakkeera and A. Saranya, "Efficient Collaborative Key Management Protocol for Secure Mobile Cloud Data Storage," in *International Conference on Intelligent Computing and Applications*, 2019, pp. 41-51.
4. M. Edwin, A. Samsudin, and S. Tan, "Implementing the honey encryption for securing public cloud data storage," in *Proceedings of First International Conference on Computer Science and Engineering*, 2017.
5. P. Li, J. Li, Z. Huang, C.-Z. Gao, W.-B. Chen, and K. Chen, "Privacy-preserving outsourced classification in cloud computing," *Cluster Computing*, vol. 21, pp. 277-286, 2018.
6. M. B. Yassein, S. Aljawarneh, E. Qawasmeh, W. Mardini, and Y. Khamayseh, "Comprehensive study of symmetric key and asymmetric key encryption algorithms," in *2017 international conference on engineering and technology (ICET)*, 2017, pp. 1-7.
7. A. Kaur, "A Review on Symmetric Key Cryptography Algorithms," *International Journal of Advanced Research in Computer Science*, vol. 8, 2017.
8. Z. Jiang, H. Jin, G. E. Suh, and Z. Zhang, "Designing Secure Cryptographic Accelerators with Information Flow Enforcement: A Case Study on AES," in *Proceedings of the 56th Annual Design Automation Conference 2019*, 2019, p. 59.
9. R. P. Adhie, Y. Hutama, A. S. Ahmar, and M. Setiawan, "Implementation cryptography data encryption standard (DES) and triple data encryption standard (3DES) method in communication system based near field communication (NFC)," in *Journal of Physics: Conference Series*, 2018, p. 012009.
10. A. R. Kumar, S. Mubeena, and V. S. Babu, "Implementation of Triple Data Encryption Standard Architecture," 2017.
11. A. Chauhan and J. Gupta, "A novel technique of cloud security based on hybrid encryption by Blowfish and MD5," in *2017 4th International Conference on Signal Processing, Computing and Control (ISPCC)*, 2017, pp. 349-355.
12. A. Jayan and B. R. Upadhyay, "RC4 in Hadoop security using MapReduce," in *2017 International Conference on Computational Intelligence in Data Science (ICCIDS)*, 2017, pp. 1-5.

13. C. Thirumalai and H. Kar, "Memory Efficient Multi Key (MEMK) generation scheme for secure transportation of sensitive data over Cloud and IoT devices," in *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*, 2017, pp. 1-6.
14. M. Thangapandiyan, P. R. Anand, and K. S. Sankaran, "Quantum Key Distribution and Cryptography Mechanisms for Cloud Data Security," in *2018 International Conference on Communication and Signal Processing (ICCSP)*, 2018, pp. 1031-1035.
15. M. Ibtihal and N. Hassan, "Homomorphic encryption as a service for outsourced images in mobile cloud computing environment," *International Journal of Cloud Applications and Computing (IJCAC)*, vol. 7, pp. 27-40, 2017.
16. A. S. Afre, M. Bharati, and S. Tamane, "DeyPos: For multi-users environments using MD5," in *2017 1st International Conference on Intelligent Systems and Information Management (ICISIM)*, 2017, pp. 251-254.
17. R. K. Vollala and L. V. Reddy, "An Improved Cryptographic Mechanism for Cloud Storage System," *International Journal of Applied Engineering Research*, vol. 12, pp. 8469-8473, 2017.
18. Y. Zhou, X. Sun, T. Mielonen, H. Li, R. Zhang, Y. Li, *et al.*, "Cirrus Cloud Optical Thickness and Effective Diameter Retrieved by MODIS: Impacts of Single Habit Assumption, 3-D Radiative Effects, and Cloud Inhomogeneity," *Journal of Geophysical Research: Atmospheres*, vol. 123, pp. 1195-1210, 2018.
19. J. Li, J. Wu, and L. Chen, "Block-secure: Blockchain based scheme for secure P2P cloud storage," *Information Sciences*, vol. 465, pp. 219-231, 2018.
20. M. A. Hossain, M. Biddut Hossain, M. Shafin Uddin, and S. Md. Imtiaz, *Performance Analysis of Different Cryptography Algorithms* vol. 6, 2016.
21. E. Agrawal and P. Pal, *A Secure and Fast Approach for Encryption and Decryption of Message Communication* vol. 7, 2017.

AUTHORS PROFILE

A.S. Kalyana Kumar *, Research Scholar, Research and Development Centre, Bharathiar University, Coimbatore, Tamil Nadu, India.

DR. T. Abdul Razak, Professor, Department of computer science and engineering, Jamal Mohamed College, Tiruchirapalli, Tamil Nadu, India.