Securing the Information using Improved Modular Encryption Standard in Cloud Computing Environment

A. Syed Ismail[1*], D. Pradeep[2], and J. Ashok[3]

[1] Department of Data Science and Business Systems, School of Computing,
SRM Institute of Science and Technology, SRM Nagar, Kattankulathur - 603 203
Chengalpattu District, Tamil Nadu, India
[e-mail: cse.ismail14@gmail.com]

[2] M.Kumarasamy college of engineering, Karur, 639113, India
[e-mail: pradeepdurai.vdr@gmail.com]

[3] Computer Science and Engineering, Annasaheb Dange college of Engineering and Technology, India
[e-mail: johnasmara984@gmail.com]

* Corresponding Author: Dr. A.Syed Ismail

Received December 6, 2022; revised March 1, 2023; accepted April 10, 2023;
published October 31, 2023

Abstract

All aspects of human life have become increasingly dependent on data in the last few decades. The development of several applications causes an enormous issue on data volume in current years. This information must be safeguarded and kept in safe locations. Massive volumes of data have been safely stored with cloud computing. This technology is developing rapidly because of its immense potentials. As a result, protecting data and the procedures to be handled from attackers has become a top priority in order to maintain its integrity, confidentiality, protection, and privacy. Therefore, it is important to implement the appropriate security measures in order to prevent security breaches and vulnerabilities. An improved version of Modular Encryption Standard (IMES) based on layered modelling of safety mechanisms is the major focus of this paper's research work. Key generation in IMES is done using a logistic map, which estimates the values of the input data. The performance analysis demonstrates that proposed work performs better than commonly used algorithms against cloud security in terms of higher performance and additional qualitative security features. The results prove that the proposed IMES has 0.015s of processing time, where existing models have 0.017s to 0.022s of processing time for a file size of 256KB.

Keywords: Cloud computing, Data Security, Improved Modular Encryption Standard, Key Generation, Logistic Map.

http://doi.org/10.3837/tiis.2023.10.013 **ISSN : 1976-7277**

1. Introduction

The use of cloud computing is on the rise. Cloud computing services are becoming popular because they are flexible, reliable, scalable, and cost-effective [1]. Using a computer network for access to share dynamically customizable resources is known as "cloud computing" [2]. Cloud computing services such as Amazon EC2 (Amazon allow users for rent to run their computer programmes and support virtual information technology (virtual IT). Amazon EC2 delivers scalable computing power in the AWS [3]. For instance, the Google App Engine is a cloud computing service that serves as a platform for hosting web applications. Examples of software as a service are Google app and Microsoft office online which offer both infrastructure and platform as a service for servers [4]. Cloud services are mainly categorized into Infrastructure as a Service (IaaS), Platform-based services (PaaS), and Software-based Services (SaaS). In which IaaS scalable and highly automated and it can be used to monitor computers, nets, storage, and other resources. IaaS enables enterprises to acquire services on-demand rather than acquiring hardware altogether [5]. PaaS, which delivers the cloud mechanism of specific apps, is the key use case for this service. Using PaaS, app creators may use a common platform to create and build their apps [6]. The most generally utilised solution for businesses in the cloud market is SaaS [7] and it might also refer to it as cloud application hosting. In SaaS, a third-party provider provides services to customers via internet. No client-side downloads or installations are required for most SaaS applications.

Cloud infrastructure is affected by a diversity of issues, with interoperability, scalability, and multi-tenancy issues. There are several vulnerabilities to cloud infrastructure, The most significant issue is protecting it from various threats as a system that uses the internet (such as embedded networks, grid computing, etc.) [8]. Cloud computing's mainstream adoption will be hampered by security issues. It is challenging to keep cloud computing services secure and safe from unauthorised access or use because of sharing these services. This is a problem that primarily affects data that has been moved to the cloud [9-10]. Network security which relates external and internal attacks on cloud computing creates safe link between cloud provider and user. There are many mechanisms and protocols in place to maintain data transfer security via the internet, and cryptography is the operative method. In cryptography, the process of converting plaintext into ciphertext is necessary. In general, it is a mechanism for transferring files securely so that only the intended receiver can access them [11]. Fig. 1 shows the workflow of securing the data in cloud.

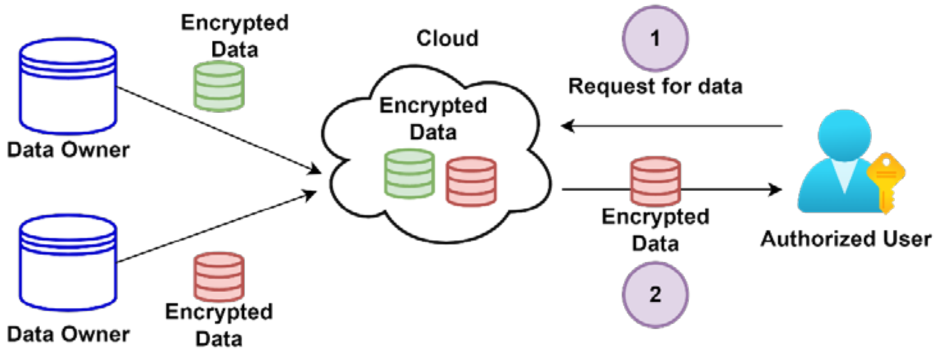

Fig. 1. Overall flow of Securing data in Cloud Computing

Unauthorized access to private data, information, or messages can be prevented by the study of cryptographic procedures [12]. Non-repudiation and data confidentiality are two of the most important features of modern cryptography, and the practise includes both aspects into its work. Cloud computing encryption is a critical subject which needs numerous studies. Encryption in cloud computing is identified based on area of encryption, according to Jaber and Bin, [13]. Security is a major problem because cloud computing maintains sensitive data and may be accessed from anywhere in the world over the internet [14]. E-commerce security, media privacy, and secure data transfer over the internet benefit by the use of cryptography [15]. These include the DES and the Rivest Shamir Adleman Algorithm as RSA [16] for data encryption and decryption. The main contribution of the research work is described as follows:

- The MES is modified in this research to provide a new, more portable encryption algorithm.
- IMES is the proposed model for strengthening cloud computing data security, where the logistic map is introduced in the key generation process.
- The proposed model is tested with other existing techniques in terms of various metrics such as modularity check, processor utilization, colligation rate and memory utilization.
- The key management is complicated in the existing techniques like blowfish. But the proposed model uses a logistic map for key generation and more mathematical expressions are used to protect the sensitive data from attackers.
- As a result, protecting data and the minimum procedures to be handled has become a top priority in order to maintain its integrity, confidentiality, protection, and privacy

The rest of the discussions are arranged in the following order. Section 2 of the paper, provides an overview of security issues. Section 3 delivers a review of relevant literature, while Section 4 clarifies the rationale for the study. Section 5 provides a description of the suggested model. It is proposed in Section 6 that the proposed model be validated using current methodologies in terms of various metrics. Section 7 concludes the work with a summary of the findings.

2. Cloud computing security impression

All the methods for safeguarding, re-establishing, and ensuring the security in computer system frameworks are incorporated into cloud security measures. Cloud application security engineering is examined in greater detail in ElasticaQ2 2015 and the Cloud Security Alliance (CSA). Data Security concerns can also be led by ISO in respect to cloud computing key security standards for a secure and efficient innovative knowledge solution. That cloud computing is extremely safe and demonstrated here. In a cloud environment, outsourcing data and software removes the user's ability to control them from the hands of the service provider. Consequently, Trust relies on the cloud's concept and the cloud's service provider for its security.

2.1. Key security necessities in Cloud

Cloud computing, like any other information expertise management system, requires a high level of confidentiality, availability, and integrity, according to NIST. As shown in **Fig. 2** further authorisation, and privacy are introduced as cloud protection standards.

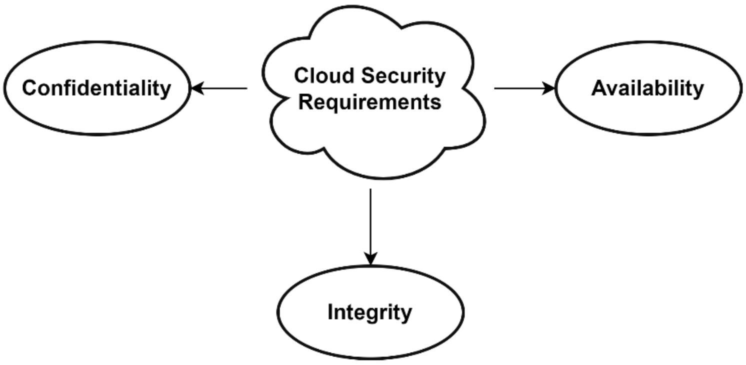

Fig. 2. Basic cloud security requirement.

- The term "confidentiality" refers to the practise of protecting client data and only allowing authorised users access to it.
- Transmitting or processing information should not modulate or modify its "integrity", and consumers should be permitted to make changes, amend, repeat, remove or remove information as they see fit.
- "Authentication" is possible to ensure the integrity of a customer's account information.
- "Availability" refers to the ease with which a customer's requested data or services can be accessed at any time and from any location.
- "Authorization" means that only those customers who have provided special information will be given access to it.

3. Related works

The hybrid technique proposed by Sajay et.al [17] aims to increase the security of cloud data by encrypting it. In order to protect and store vast amounts of data in the cloud, encryption methods are commonly employed. In order to improve cloud security, this research uses a combination of homographic and blowfish encryption. If the security issues are addressed, cloud storage solutions for both small and large businesses will be the future. On the subject of cloud computing security, it is still in its infancy when it comes to protecting consumers' personal information, which is provided by Tajammul et.al [18]. Since the dawn of cryptography, a number of algorithms have been studied and used. However, none of these methods generate the encryption key on their own. This study developed an algorithm that generates a key based on the input and then encrypts the data using the key generated. Encrypted statistics will be loaded to the cloud, while the decryption key will remain on the local server.

An effective time-variant attribute-based multitype (TAM) encryption method was devised by Kumaresan et.al [19]. The TAM procedure maintains a nomenclature of properties and related keys for encryption and decryption. The ciphertext was generated using the corresponding keys. For better integrity organization and security performance than earlier algorithms, this new taxonomy's content has been constantly changing in each time window. Up to 89.6% greater security performance is provided by the TAM method. Using this technique, you can also cut the time difficulty by up to 21 seconds and boost the throughput by up to 96%.

An ABE technique based on the SM9 encryption algorithm (SM9-ABE) was devised by Ji et.al [20] to make SM9 allow control that would be improved and implemented in DCC. In the selective CPA model under the DBDH supposition, our suggested technique (SM9-ABE) has been demonstrated to secure extremely. In addition, we put SM9-ABE into action and access its usability in the real world. The implementation shows that our strategy achieves well in terms of security and functionality, and that the additional time cost is tolerable. Viswanath et.al [21] has created a system that prevents insider attacks. For the purpose of protecting huge data before it is stored in many clouds, an encryption method known as a hybrid was created. Storage environments are used for the simulation investigation. Using suggested approach, the encryption procedure was recorded at a rate of approximately 2630 KB/S. The findings display the suggested method outperforms the benchmark procedures in terms of performance.

The multilevel security system described by Thabit et.al [22] is more secure than any existing single-level encryption procedure. Additionally, our algorithm in several instructions, such as when execution uploads and downloads of a given file, thanks to the proposed technique's ability to limit access to cloud data to pre-authorized users only. As a result of Vidhya et.al [23], the work seeks to develop an advanced encryption algorithm (FAEA) that is cost-effective and satisfiable for cloud-based Big Data use. The efficiency, scalability, and security of the FAEA approach have been analysed, and it has been found to be 98% better than the currently used HDFS and Map Reduce Encryption Scheme methods (MRE). There are many issues that must be addressed when using Big Data on the Cloud, and our study seeks to address such issues. For data storage in a multi-cloud context, Seth et.al [24] use three basic algorithms: for user identity, Data encryption is done via Blowfish, and RSA is used to convey the secret key for Blowfish's data encryption. The proposed solution is evaluated based on the time it takes to download, upload, and access it. In order to safeguard the secrecy of the data, this architecture allows secure, quick, and lightweight access to data. Using client-side data encryption, Sohal et.al [25] developed a novel cryptographic technique for encrypting data before uploading it to the cloud. Symmetric-key cryptography using DNA is the basis for this technology. In addition to providing our approach's comprehensive design, we also compared it to the already available symmetric-key methods.

Etemad et al., [26] strengthen the cloud security by proposing dynamic data outsourcing scheme. The integrity verification process was done by black-box access to implicitly authenticated data. A homomorphic verifiable tag scheme is designed for blockless verification. Yang, et al., [27] developed a model for inner product computation using Inner Product Functional encryption (IPFE). There are two privacy weakness presented in the model such as master secret key and encrypted vector. To resolve this weakness, the study designed the strengthened IPFE. The research work also constructs the scheme for outsourced computation scheme for inner product and uses the computation cost and storage overhead as vector sizes.

Yang et al., [28] considered the IPFE by designing the verifiable computation scheme, where the secret key is transformed into blinded form to preserve the key privacy. The computation overhead is eliminated by performing the authentication procedures of cloud servers before the computation. Therefore, only the allowed user can use the outsourced data that are shared in the cloud computing. Yang, et al., [29] resolved the weakness of unbounded-size database by designing verifiable update scheme. Initially, authenticated matrix commitment is developed and give the access to the scheme. To assure the ownership of the opened data, the ordered data collection is represented in the matrix format. The validation provided that the model is efficient than existing techniques.

3.1. Problem statement with solution

As can be seen from the results, our suggested technique beats more standard ones in terms of ciphertext size, time, and throughput. It is thus more efficient and provides higher performance with the new technique provided. All the existing techniques uses blowfish, AES, DES, RSA or time-variant based attribute algorithms for encryption and no mathematical encryption techniques are considered for securing the data. AES is less secure and uses 256-bit key for sharing the sensitive data, where blowfish algorithm requires a unique key for users, so key management becomes complicated, when a greater number of users are added in the network. But the proposed model uses a logistic map for key generation and more mathematical expressions are used to protect the sensitive data from attackers.

4. Proposed System

A certain set of requirements must be met in order for an encryption algorithm to be considered secure. Based on the present literature, the following criteria have been established for the newly created algorithm:

- A full character set should be encrypted, and each plain-text message should be encoded into a special order.
- The encryption method must also be sophisticated.

Fig. 3 provides the workflow of the proposed model.

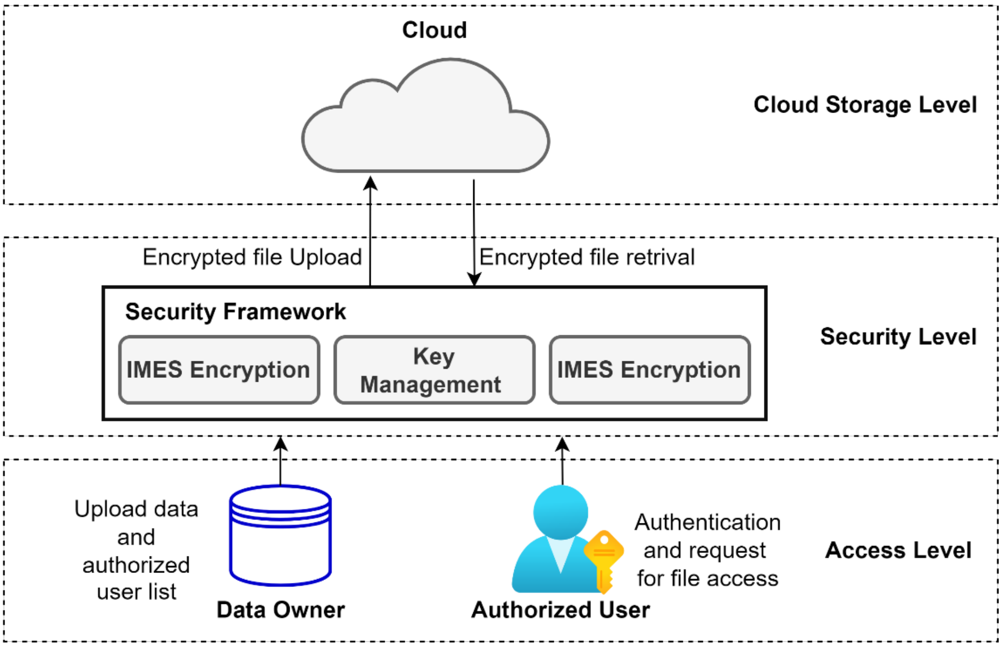

Fig. 3. Securing Data by using proposed model

The used representation in the proposed model is depicted in **Table 1**.

Table 1. The used notations.

Plaintext	PT
Key	K
Lightly Encrypted Plaintext	LEPT
Expansion	Exp
Discard Expansion	DExp
Permutation for r^{th} round	Pr
Substitution for r^{th} round	Sr
Extension	Ext
Contraction	Cn
Left half key for any r^{th} round	K_L^r
Right half key for any r^{th} round	K_R^r

In Eq. (1), block cipher (BC) is described in y-bit key.

$$(PT, K) \to CT \tag{1}$$

$$CT = \varepsilon(PT, K) \tag{2}$$

Encryption conditions are shown in Equations (1) and (2) using the private key that is obtained by a logistic map.

4.1. Key generation block

When it comes to encryption and decryption, the key is the most crucial component. The security of the data is jeopardised if an attacker is able to decipher this key. The integrity of the data is entirely dependent on this key. Different actions are approved out to avert the possibility of a weak key as well as to improve the key's strength in order to produce confusion and dispersion. A logistic map is utilised to generate the key that is employed in the diffusion process. The logistical map is distinct by the following:

$$Y_{n+1} = aY_n(1 - Y_n) \tag{3}$$

Assume that the control structure has an input range of zero through four and that output sequence Yn has an input range of zero through one. When $a \in [2.57, 4]$occurs, the map becomes chaotic. The following are a list of the most important steps in the creation process:

1. Use the following equation to determine the beginning value of the logistic map, which is dependent on the plain text PT:

$$Y_0 = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} PT(i,j)}{M\times N\times 255} \tag{4}$$

The integers M and N relate to the sum of rows and columns in the plaintext, correspondingly,

2. Iterate and skip the first N_0 entries for a novel sequence S of size MN.
3. The subsequent formula to Calculate the key:

$$K = mod(floor(S(i) \times 10^{14}), 256), i = 1{:}MN \tag{5}$$

4.2. Mathematical Description for Encryption

The Equation (6) describes plaintext expansion from 56-bit to 64-bit. So-called Lightly Encrypted Plaintext is the result of this process.

$$LEPT = Ext(PT) \tag{6}$$

LEPT is provisionally expanded from 64 to 128 bits in order to execute Key whitening, as indicated in Equation (7). Key whitening is done with the K0 key. The LEPT can only be transformed once with the K0 key. In contrast, from K1 to K9, each key converts the LEPT twice.

$$(LEPT)Exp \,\overline{\oplus}\, K_0 \tag{7}$$

$DExp$ is used to lower the LEPT to 64 bits in order to remove the transient contraction and it is shown in (8).

$$DExp((LEPT)Exp) \oplus K_0 \tag{8}$$

Afterwards, the Permutation based on Equation (9) is carried out as follows.

$$(LEPT\ \overline{\oplus}\, K_0)P^r \tag{9}$$

For any r-th round, the replacement is followed by the addition of the left half key and the subtraction of the right half key, which is given in Equation 10.

$$\left(\left((LEPT\ \overline{\oplus}\, K_0)P^r\right) S^r\, \overline{\oplus}\, K_L^r\right) \overline{\oplus}\, K_R^r \tag{10}$$

4.3. Decryption of the Algorithm

To access the encrypted data from the cloud, the cloud computing's user would first need to decrypt it. Equation (11) describes how key subtraction, which takes place on the encryption side, is cancelled out at the decryption side by subtracting the right half key of the rth round.

$$\left(\left((LEPT\ \overline{\oplus}\, K_0)P^r\right) S^r\, \overline{\oplus}\, K_L^r\right) \overline{\oplus}\, K_R^r\, \overline{\oplus}\, K_R^r \tag{11}$$

The next step is to conduct key addition to cancel the effect of key adding on the encryption side, as explained in Equation (12) that utilising the left half key for any subsequent rounds.

$$\left(\left((LEPT\ \overline{\oplus}\, K_0)P^r\right) S^r\, \overline{\oplus}\, K_L^r\right) \overline{\oplus}\, K_L^r \tag{12}$$

Any subsequent rounds of inverse substitution are used to eliminate the effect of substitution on the encryption side, which is shown in Eq. (13).

$$S_r'(\left((LEPT\ \overline{\oplus}\, K_0)P^r\right) S^r) \tag{13}$$

According to Equation (14), the permutation effect on the encryption side is cancelled using inverse permutation for any rth round.

$$P_r'\left((LEPT\ \overline{\oplus}\, K_0)P^r\right) \tag{14}$$

Key decryption side would counteract effect on the encryption side, which is provided in Eq. (15).

$$LEPTK_0\, \overline{\oplus}\, K_0 \tag{15}$$

When decryption is complete, LEPT is transformed into standard PT as described in Equation (16).

$$PT = Cn(LEPT) \tag{16}$$

As a result, as previously explained, the plaintext is obtained during the decryption process. **Fig. 4** is a flowchart of the full IMES system.

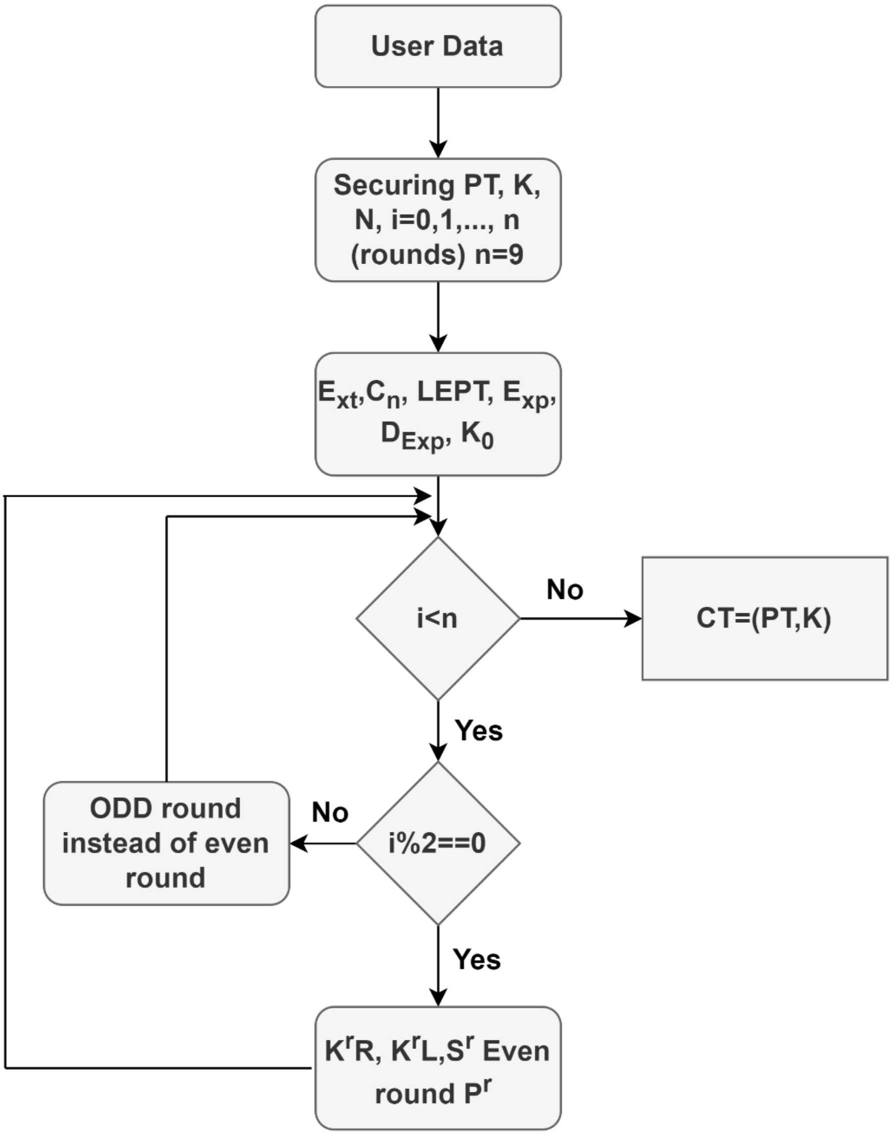
User Data
Securing PT, K, N, i=0,1,..., n (rounds) n=9
Ext,Cn, LEPT, Exp, DExp, K0
i<n
No
CT=(PT,K)
Yes
i%2==0
No
ODD round instead of even round
Yes
KrR, KrL,Sr Even round Pr

Fig. 4. Flowchart for data enciphering.

Algorithm 1: Algorithm-IMES Encryption

```
1 $\boldsymbol{Declaration}$;
2 $PT \leftarrow Plaintext$;
3 $N \leftarrow Size\ of\ PT$;
4 $CT \leftarrow Ciphertext$;
5 $PPT \leftarrow Padded\ form\ of\ PT$;
6 $BPT \leftarrow Binary\ form\ of\ PT$;
```

```
$EPT \leftarrow Extended\ form\ of\ PT$;
$AGK \leftarrow Key\ generation\ using\ logistic\ map$;
$Ki \leftarrow Key, i = 0,1,..,9$;
$CT \leftarrow Ciphertext$;
$\boldsymbol{Execution - Encryption}$;
$N \leftarrow Length(PT)$;
$BPT \leftarrow BinaryFormatting(PT)$;
$\boldsymbol{if}\ N < 64\ \boldsymbol{then}$
    $PPT \leftarrow Padding(BPT)$;
    $EPT \leftarrow Extension(PPT)$;
$KEY \leftarrow Transformation(AGK)$;
$Key \leftarrow Whitening(K0)$;
$\boldsymbol{while}\ i < 9\ \boldsymbol{do}$
    $Permutation(EPT)$;
    $CT \leftarrow Shifting(EPT)$;
    $CT \leftarrow Substitution(EPT)$;
    $CT \leftarrow Key - Addition(Ki)$;
    $CT \leftarrow Key - subtraction(Ki)$;
    $i + +$;
$\boldsymbol{end}$
$KeyEncryption(AGK)$
```

Algorithm 2: Algorithm-IMES Decryption

```
$\boldsymbol{Declaration}$;
$PT \leftarrow Plaintext$;
$CT \leftarrow Ciphertext$;
$N \leftarrow Size\ of\ CT$ ;
$PPT \leftarrow Padded\ form\ of\ PT$;
```

```
$UPCT \leftarrow Un - Padded\ form\ of\ CT$;
$SCT \leftarrow String\ form\ of\ CT$;
$CCT \leftarrow Contracted\ form\ of\ PT$;
$AGK \leftarrow Key\ Generation\ using\ logistic\ map$;
$Ki \leftarrow Key, i = 0, 1, \ldots 9$;
$PT \leftarrow Plain\ Text$;
$\boldsymbol{Execution - Decryption}$;
$Key - Decryption(AGK)$;
$KEY - Transformation(AGK)$;
$\boldsymbol{while}\ i < 9\ \boldsymbol{do}$
    $PT \leftarrow Permutation(EPT)$;
    $PT \leftarrow Shifting(EPT)$;
    $PT \leftarrow Substitution(EPT)$;
    $PT \leftarrow Key - Addition(Ki)$;
    $PT \leftarrow Key - Substraction(Ki)$;
    $PT \leftarrow i - Key - Whitening(K0)$;
    $i++$;
$\boldsymbol{end}$
$SPT \leftarrow StringFormate(CT)$;
$\boldsymbol{if}\ PPT! = NULL\ \boldsymbol{then}$
    $UPCT \leftarrow Un - padding(SPT)$;
    $CCT \leftarrow Contraction(UPCT)$;
    $N \leftarrow Length(PT)$;
```

5. Results and Discussion

To conduct the proposed scheme's performance analysis, the following environmental factors were taken into consideration that is shown in Table 2.

Table 2. Experiments Setup.

Setup	Description
OS	Windows 10
Processor	Intel Core; CPU @ 2.90GHz
System	X-64 based Processor, 64-bit OS
Platform	Visual C++

5.1. Check for Modularity

Table 3 shows the module-based processing use of IMES with various input sizes. In addition, the IMES's module-based execution time is listed below.

Table 3. Rate of utilization processor.

	Modular-Analysis	Time (1KB sec)	Time (2KBs sec)	Time (3KBs sec)
1	Key-whitening	0.000007030	0.000019749	0.000001057
2	Rounds	0.000371302	0.000035610	0.000025030
3	Key-Transformation	0.0620079	0.00341145	0.00275324
4	Key Encryption	0.000394236	0.000004572	0.000003534

The elapsed time estimation for IMES encryption has been completed. In cryptography, the enciphering time measures how long it takes to convert raw data into ciphertext. The throughput of any algorithm can be estimated using the encryption time. It has an impact on encryption's speed. The greater the throughput, the lower the power consumption that will be shown in the following, different outcomes were obtained by varying the input size. **Table 4** and **Fig 5** and **Fig 6** provides the investigation of IMES in terms of key transformation.

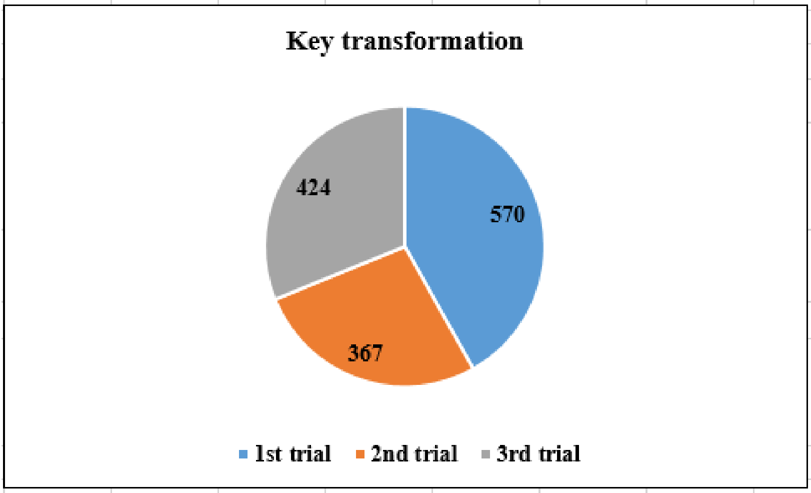

Fig. 5. Results of proposed IMES in terms of key transformation.

Table 4. Investigation of IMES in terms of modularity check

	1st trial	2nd trial	3rd trial
Key transformation	570	367	424
Key whitening	67	43	28
Rounds	87	48	39
Key encryption	96	56	43

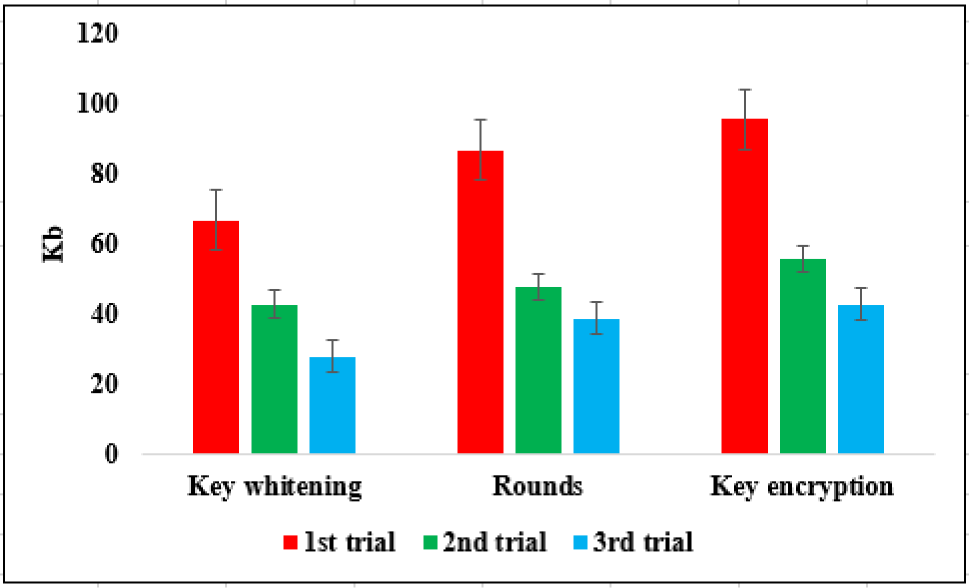

Fig. 6. Results of proposed IMES for various trails on modular investigations

5.2. Performance on Processor Utilization

Here, the validation of CPU utilization is compared for proposed IMES with existing models in terms of various trials, which is provided in **Table 5** and **Fig 7** shows how MES compares to other cryptographic algorithms in terms of performance when it comes to CPU use. The existing techniques such as AES [20], Blowfish [17,24] and DES [19] are also considered for this comparison, however these techniques are all implemented with the data used in this research work.

Table 5. Experimental Analysis on Processor utilization

Different Algorithms	1st trial	2nd trial	3rd trial	4th trial
Proposed IMES	401	156	103	357
AES	460	869	956	986
Blowfish	480	990	998	1010
DES	520	230	303	340
RC5	1530	660	740	980
3DES	410	180	300	301

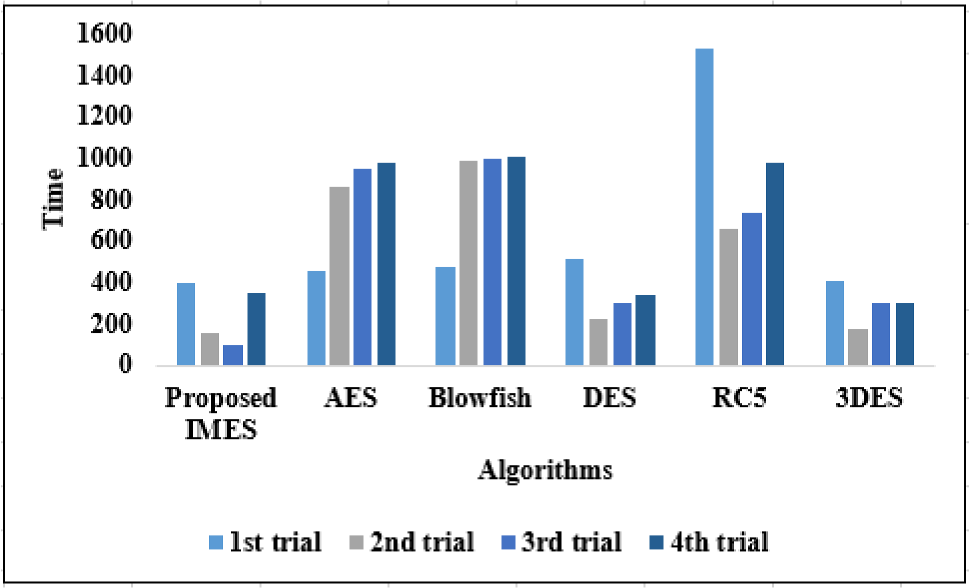

Fig. 7. Processor Utilization of proposed model with existing models

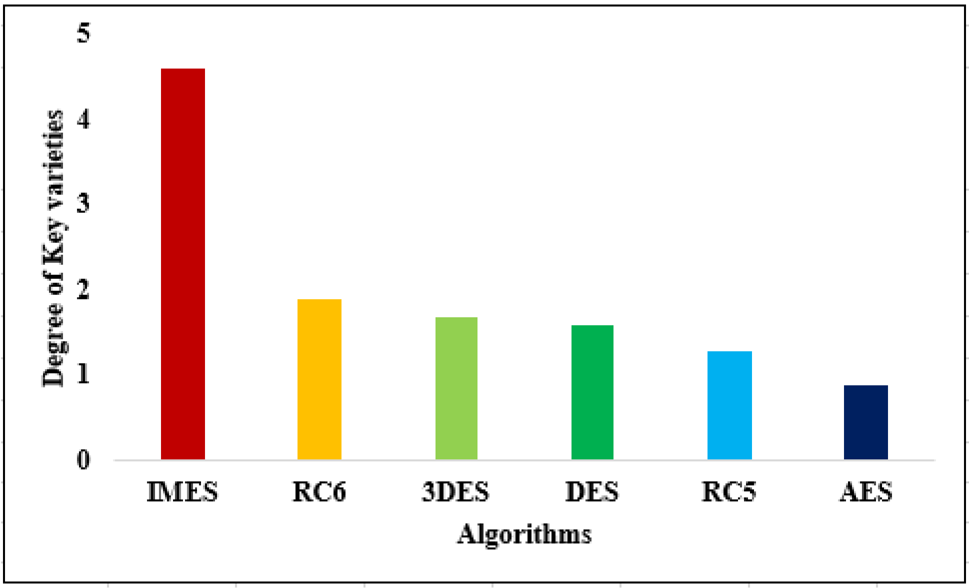

Fig. 8. Graphical Representation of proposed model in terms of Degree of Key varieties.

Different input sizes were used to examine the processor time for each round. MES's key transformation uses a larger percentage of CPU time than the other modules. Block cyphers like MES and HI are evaluated against the MCC environment's security in this section. Analyzing how long a CPU spends on a given computation is known as processor utilisation. It shows the processor's current workload. In general, the more CPU time spent encoding, the more work the processor must do. An investigation of how varied input sizes and platforms affect processor time estimation is carried out in these trials. **Fig. 8** depicts the various key schemes and the key types requirements for different algorithms.

Qualitative comparison is used in this study. One type of key is supported by DES, RC5, RC6, Blowfish, and IDEA, while AES provides three types of keys, and MES provides five types of keys. According to the graph below, MES has the highest level of important variances. **Table 6** and **Fig. 9** shows the performance analysis of key colligation rate.

Table 6. Key-data colligation rate.

Encryption scheme	IMSE	AES	Blowfish	DES	RC5	3DES
Single round key colligation	2.01	1.04	1.02	1.03	1.02	1.04

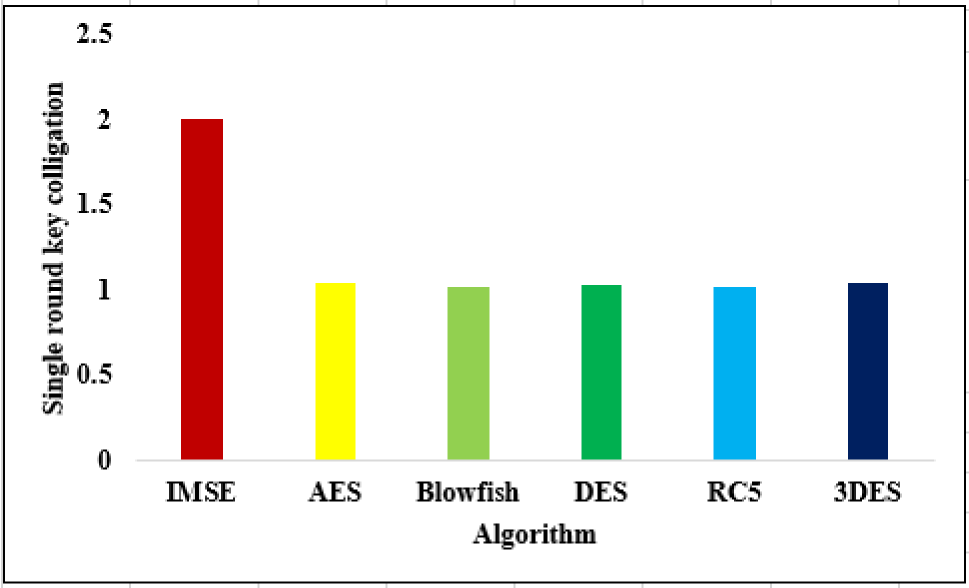

Fig. 9. Colligation rate of key for proposed model

Each key transforms data twice with each round, except stage. Key whitening (KW) is only one of subsuming metrics, as key subtraction and key adding are also important. In the single round, the key colligation rate is 2.01 for proposed model, where the existing techniques such as AES, Blowfish, DES, RC5 and 3DES achieved 1.04, 1.03 and 1.04 rate of key colligation.

5.3. Analysis of Time Complexities

IMES is a block-size-dependent algorithm like AES, DES, 3DES, and so on. Regardless of the input size, this algorithm has a temporal complexity as($O(1)$). IMES has a spatial complexity of $O\ (n)$.Furthermore, IMES outperforms other commonly used algorithms in terms of memory and CPU use as well as key variances and data collecting rates. This makes IMES a better option because of its low memory and CPU utilisation. **Fig. 10** and **Table 7** shows the experimental analysis.

Table 7. Memory utilization.

Encryption scheme	IMSE	AES	Blowfish	DES	RC5	3DES
Memory utilization (KB)	984	1357	1293	1785	1624	1896

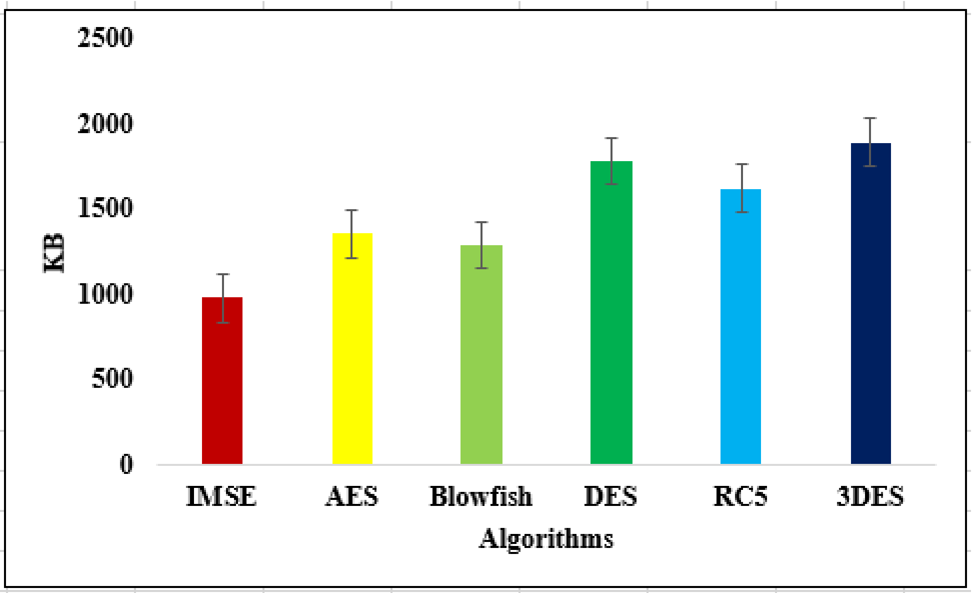

Fig. 10. Memory Utilization

Table 8 and **Table 9** provides the experimental analysis of processing time, encryption time, and decryption time for various file sizes by using different algorithms.

Table 8. Comparison in terms of processing time (s).

Key Size	DES	AES	Blow fish	3DES	IMES
1MB	0.13	0.13	0.13	0.09	0.06
10MB	0.32	0.32	0.26	0.87	0.64
50MB	1.89	1.61	2.13	4.5	2.9
70KB	156	256	256	192	64
Possible key	2^{56}	2^{256}	2^{256}	2^192	2^{64}
File 256KB	0.017	0.017	0.017	0.022	0.015
Size 512KB	0.051	0.051	0.051	0.046	0.03

When the key size is less than 20MB, the DES, AES and Blowfish has 0.13s, 3DES has 0.09s and proposed model has only 0.06s for 1MB key size. At the same time, when the key size is 70MB, the proposed model has 64s of processing time, AES has 256s, DES has 156s, Blowfish has 256s and 3DES has 192s. When the file size is 512KB, the proposed model has 0.03s, the DES, AES and Blowfish has 0.051s. From this analysis, it is clearly proving that proposed IMES achieved better performance than existing techniques.

Table 9. The table shows the response time in seconds.

	File Size	Encryption time	Decryption time	Total Time
DES	1.2 MB	**6.618632**	**0.681449**	**7.300081**
	2.9 MB	**12.194479**	**1.134509**	**13.328988**
	7.2 MB	**31.071637**	**2.842029**	**33.913666**
3DES	1.2 MB	12.547342	3.537952	16.085294
	2.9 MB	24.56739	4.748292	29.315682
	7.2 MB	59.83622	5.738291	65.574511
AES	1.2 MB	4.537732	0.782323	5.320055
	2.9 MB	10.53943	1.649226	12.188656
	7.2 MB	29.64821	2.649457	32.297667
Blow fish	1.2 MB	6.649212	0.679712	7.328924
	2.9 MB	11.93644	1.873523	13.809963
	7.2 MB	33.74941	2.847364	36.596774
MSE	1.2 MB	10.74930	3.972811	14.722111
	2.9 MB	23.34562	4.378021	27.723641
	7.2 MB	58.45631	6.01352	64.46983

When the file size is 2.9MB, DES has 12.19s of encryption time, 1.13s of decryption, AES has 10.53s of encryption time, 1.64s of decryption time, where the proposed model has 23.34s for encryption, 4.37s for decryption and this is due to logistic map for key generation. The other existing techniques are randomly used the keys for further process and these effects made the proposed model has high encryption and decryption time than existing models. But, the efficiency of IMES is satisfactory than other models in terms of different parameters.

7. Conclusion

Cloud computing security has climbed to the top of cloud customers' concerns. Cryptography is the most effective of several ways and strategies. In this study, a novel, lightweight cryptographic method called IMES has been proposed to increase cloud computing security. Encryption relies on symmetric cryptography. For the purpose of benchmarking the suggested method's performance against other well-known cryptographic algorithms, we looked at parameters including block size, key length, prospective key, mathematical processes, cipher type, and security power. There is a clear development in encryption and decryption with the IMES approach, according to experimental data. This method provides great security and minimal computational costs. In terms of data collection and processing speed, cloud computing is even more effective. It is possible to further enhance the suggested work's efficiency by incorporating quantum computing in order to make it more user-friendly in terms of privacy and security issues.

Data Availability Statement

The authors would like to hereby state that, “**Data sharing not applicable to this article as no datasets were generated or analysed during the current study**”.

Acknowledgments

Not Applicable

Research funding

None declared.

Author contributions

The author has accepted responsibility for the entire content of this manuscript and approved its submission.

Competing interests

Author state no conflict of interest.

References

[1] I.Ahmed, “A brief review: security issues in cloud computing and their solutions,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol.17, no.6, pp.2812-2817, Dec 2019. Article(CrossRef Link)

[2] Y. Geng, “Homomorphic encryption technology for cloud computing,” *Procedia Computer Science*, vol.154, pp.73-83, Dec 2019. Article(CrossRef Link)

[3] E. Sivaraman and R. Manickachezian, “Efficient multimedia content storage and allocation in multidimensional cloud computing resources,” *International Journal of Intelligent Systems Technologies and Applications*, vol.18, no.1/2, pp.20-33, Feb 2019. Article(CrossRef Link)

[4] Pradeep Kumar Tiwari, K. Kannan, Duggineni Veeraiah, Nikhil Ranjan, Jain Singh, Ghalib H. Alshammri, Awal Halifa, "Security Protection Mechanism in Cloud Computing Authorization Model Using Machine Learning Techniques," *Wireless Communications and Mobile Computing*, vol. 2022, pp.1-12, July 2022, Article ID 1907511. Article(CrossRef Link).

[5] S.Eswaran, D.Dominic, J.Natarajan and P.B.Honnavalli P. B, “Augmented intelligent water drops optimisation model for virtual machine placement in cloud environment,” *IET Networks*, vol.9, no.5, pp.215–222, April 2020. Article(CrossRef Link)

[6] J.Samuel Manoharan, “A Novel user layer cloud security model based on chaotic Arnold Transformation using fingerprint biometric traits,” *Journal of Innovative Information Processing*, Vol.3, no.1, pp.36 – 51, April 2021. Article(CrossRef Link)

[7] U.A.Butt, R.Amin and M.Mehmood, “Cloud Security Threats and Solutions: A Survey,” *Wireless Pers Commun*, vol.128, pp.387-413,Sep 2023. Article(CrossRef Link)

[8] S.Eswaran, and M.Rajakannu, “Multiservice Load Balancing with Hybrid Particle Swarm Optimization in Cloud-Based Multimedia Storage System with QoS Provision,” *Mobile Networks and Applications*, vol.22, no.4, pp.760–770, Sep 2017. Article(CrossRef Link)

[9] G.S. Pavithra and N.V. Babu, “Energy efficient hierarchical clustering using HACOPSO in wireless sensor networks,” *Int. J. Innovat. Technol. Explor. Eng*., vol.8, no.12, pp. 5219-5225, Oct 2019. Article(CrossRef Link)

[10] S. Singh, Y.S. Jeong and J.H. Park, “A survey on cloud computing security: issues, threats, and solutions,” *Journal of Networks and computer applications*, vol.75, pp.200-222, Nov 2016. Article(CrossRef Link)

[11] E.Sivaraman and R.Manickachezian, “Intelligent decision-making service framework based on analytic hierarchy process in cloud environment,” *International Journal of Networking and Virtual Organisations*, vol.21, no.2, pp.221-236, Aug 2019. Article(CrossRef Link)

[12] G.Prabu Kanna and V.Vasudevan, “A fully homomorphic–elliptic curve cryptography based encryption algorithm for ensuring the privacy preservation of the cloud data,” *Cluster Computing*, vol.22, no.4, pp.9561-9569, July 2019. Article(CrossRef Link)

[13] R.Adee and H.Mouratidis, “A Dynamic Four-Step Data Security Model for Data in Cloud Computing Based on Cryptography and Steganography,” *Sensors (Basel)*, vol.22, no.3, pp.1-12, Feb 2022. Article(CrossRef Link).

[14] E.K.Subramanian and L.Tamilselvan, “Elliptic curve Diffie–Hellman cryptosystem in big data cloud security,” *Cluster Computing*, vol.23, no.4, pp.3057-3067, Dec 2020. Article(CrossRef Link)

[15] E.Sivaraman and R.Manickachezian, “Unevenness measurement using the support vector machine and dynamic multiservice load balancing with modified genetic algorithm in cloud-based multimedia system,” *International Journal of Computer Aided Engineering and Technology*, vol.10, no.6, pp.732-747, Oct 2018. Article(CrossRef Link)

[16] R. Kumar and R. Goyal, “On cloud security requirements, threats, vulnerabilities and countermeasures: A survey,” *Comput. Sci. Rev.*, vol. 33, pp. 1–48, Aug 2019. Article(CrossRefLink)

[17] M.Tajammul and R.Parveen R, “Auto encryption algorithm for uploading data on cloud storage,” *International Journal of Information Technology*, vol.12, no.3, pp.831-837, Feb 2020. Article(CrossRef Link)

[18] Hossein Abroshan, “A Hybrid Encryption Solution to Improve Cloud Computing Security using Symmetric and Asymmetric Cryptography Algorithms,” *International Journal of Advanced Computer Science and Applications*, vol.12, no.6, pp.1-7, Apr 2021. Article(CrossRef Link)

[19] Kumaresan S. and Shanmugam V, “Time-variant attribute-based multitype encryption algorithm for improved cloud data security using user profile,” *The Journal of Supercomputing*, vol.76, no.8, pp.6094-6112, Aug 2020. Article(CrossRef Link)

[20] Ji H, Zhang H, Shao L and Luo M, “An efficient attribute-based encryption scheme based on SM9 encryption algorithm for dispatching and control cloud,” *Connection Science*, vol.33, no.4, pp.1094-1115, Nov 2021. Article(CrossRef Link)

[21] Viswanath G. and Krishna P.V, “Hybrid encryption framework for securing big data storage in multi-cloud environment,” *Evolutionary Intelligence*, vol.14, no.2, pp.691-698, June 2021. Article(CrossRef Link)

[22] F.Thabit, S.Alhomdy and S.Jagtap, “Security analysis and performance evaluation of a new lightweight cryptographic algorithm for cloud computing,” *Global Transitions Proceedings*, vol.2, no.1, pp.100–110, June 2021. Article(CrossRef Link)

[23] A.Vidhya and P.M.Kumar, “Fusion-based advanced encryption algorithm for enhancing the security of Big Data in Cloud,” *Concurrent Engineering*, vol.30, no.2, pp.171-180, May 2022. Article(CrossRef Link)

[24] B.Seth, S.Dalal, V.Jaglan, D.Le, S.Mohan and G.Srivastava, “Integrating encryption techniques for secure data storage in the cloud,” *Transactions on Emerging Telecommunications Technologies*, pp.1-24, Sep 2020. Article(CrossRef Link)

[25] M.Sohal and S.Sharma, “BDNA-A DNA inspired symmetric key cryptographic technique to secure cloud computing,” *Journal of King Saud University-Computer and Information Sciences*, vol.34, no.1, pp.1417-1425, Jan 2022. Article(CrossRef Link)

[26] M Etemad, and A Küpçü, “Generic dynamic data outsourcing framework for integrity verification,” *ACM Computing Surveys (CSUR)*, vol.53, no.1, pp.1-32, Feb 2020. Article (CrossRef Link)

[27] H Yang, Y Su, and J Qin, “Privacy-preserving outsourced inner product computation on encrypted database,” *IEEE Transactions on Dependable and Secure Computing*, vol.19, no.2, pp. 1320-1337, Mar 2022. Article(CrossRef Link)

[28] H Yang, Y Su, and J Qin, “Verifiable inner product computation on outsourced database for authenticated multi-user data sharing,” *Information Sciences*, 539, 295-311, Oct 2020. Article (CrossRef Link)

[29] H Yang, D Feng, and J Qin, "Efficient Verifiable Unbounded-Size Database From Authenticated Matrix Commitment," *IEEE Transactions on Dependable and Secure Computing*, pp.1-17, Nov 2022. Article(CrossRef Link)

Dr. A. Syed Ismail (Abdul Lathif Syed Ismail) obtained his Bachelor’s degree in Information Technology from Anna University. Then he obtained his Master’s degree in Computer Science and Engineering from Anna University and PhD in Computer Science majoring in Cloud Computing, Cloud Security and Networking both from Vellore Institute of Technology. He has also obtained CCNA professional qualifications. Currently, he is an Assistant Professor at the Faculty of Computer Science and Engineering, SRM Institute of Science and Technology. His specializations include Cloud network, WSN, Networking and Cloud Security. His current research interests are ,WSN, Networking, Network Security, Cloud Security, Fuzzy system and Image Processing.

Dr. Pradeep Duraisamy is currently working as an Associate Professor of Computer Science and Engineering, M.Kumarasamy College of Engineering, Karur, Tamilnadu, India. He has received his Ph.D from Anna University, Chennai, Tamilnadu, India. He has around 12 years of academic experience in engineering colleges. He has published around 30 research articles in various Journals, Conferences, Book Chapters and Patents. He has organized more than 30 events for the betterment of student and faculty communities. He is the life member of ISTE and IAENG. His area of interest includes Cyber Security, Artificial Intelligence, Data Mining, Big Data, Cloud Computing and IoT. He is contributing as a reviewer of various reputed journals. He can be contacted at email: pradeepdurai.vdr@gmail.com.

Dr. Ashok. J received his B.E. Degree in Electrical and Electronics Engineering from the University of Madras, in 2003, M.E. Degree in Applied electronics from Anna University, Chennai, in 2005, and Ph.D. Degree in Applied Electronics from PRIST University, India, in 2016. He has teaching experience of more than 15 years in the areas of Microwave Engineering, Remote sensing and GIS, Digital Telephony, Networks, Micro-processor, Neural Networks, and Electrical and Digital Circuits. Liner integrated circuits, Microprocessor and microcontroller, Medical Electronics, embedded system. Prof. Ashok. J currently serves as an Associate Professor of the Department of Electronics and Communication Engineering, V.S.B. Engineering College Karur, India. He was an Associate professor at the College of Engineering & Technology MAI NEFHI, STATE OF ERITREA in the year 2019-2021 in the Department of Electronics and also a Professor at Annasaheb Dange College of Engineering & Technology, Sangli, and Maharashtra, India in the Department of Electronics and Telecommunication in the year of 2017-2019.