International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V73I4P127

Volume 73 Issue 4, 322-340, April 2025

Original Article

Modern Lightweight Cryptography for Secured Cloud Data Using Constrained Spherical Scyphozoan Jellyfish Optimizer

T. Rathi Devi[1*], S. Nallusamy[1], D. Sobya[2], P.S. Chakraborty[1], P. Divya[3]

[1]*Department of Adult and Continuing Education and Extension, Jadavpur University, Kolkata, India.*
[2]*Department of Computer Science & Engineering, Institute of Engineering and Management, University of Engineering and Management, Kolkata, India.*
[3]*Department of Computer Science & Engineering, Dr. M.G.R. Educational and Research Institute, Chennai, Tamilnadu, India.*

[*]*Corresponding Author: revathi.rathi26@gmail.com*

Received: 15 March 2025 Revised: 19 March 2025 Accepted: 20 March 2025 Published: 26 April 2025

Abstract *- Shared data storage and processing resources are made exclusively available through cloud computing, an internet-based computing model. Processing the gathered data on different cloud providers that have limited computing power has become more and more necessary. The realization of the algorithms for encrypting data in cloud computing must be homomorphic and lightweight, as they cannot handle large amounts of processing. The primary disadvantage is that it requires users and businesses to grant third parties access to their confidential information. A new area of cryptographic study called Lightweight Cryptography (LWC) framework with the generation of an optimized key using Constrained Spherical Scyphozoan Jellyfish-based Dynamic Key (CS2JDK) provides Cryptographic security to optimize the process to ensure security. Where the LWC uses less computational resources like less power consumption, less memory allocation, etc. Controlling the exploration and exploitation search to prevent issues with harmonic convergence or being caught in local optima is its most difficult assignment. The goal is to enhance the ability to be confined into local optima by introducing a new orthogonal learning-based variation of the search optimizer. Integrated encryption routing involves the transfer of compressed data that has been added by this key. Next, data aggregation based on compression is used to lower the data size and, therefore, the transmission cost. The data size reduction can be done by utilizing the enriched Principal Component Analysis (PCA)-based compressing scheme before transmitting data. The compressed data set is much more efficient to transmit. It converts the high-dimensional dataset to the low-dimensional data set. A generated key produced is affixed to the data that the device transmits. The main emphasis of this work is to investigate guaranteed cloud data security and outperform other relevant systems on cloud data using new optimizing techniques.*

Keywords *- Cloud computing, Lightweight cryptography, Homomorphic encryption, Jellyfish optimizer, Dynamic key generation.*

1. Introduction

Over the past few decades, there has been a considerable increase in internet-based user communication. Furthermore, increasing customer numbers have started utilizing data. The sharing of sensitive and private information has become more transparent. Sensitive data security is necessary for secure transformation. One location where private data can be kept is in the cloud; however, data security is an issue that needs to be resolved. Technology and techniques for distributed computing have evolved significantly in the last few years. Numerous architectures, distributed network prototypes, and their infrastructures, like pervasive, autonomous, cloud, and so on, have been developed as a result of this expansion. A network of computers, typically connected via the internet, that shares a distributed quantity of services offered to satisfy customer needs is referred to as Cloud Computing (CC) [1]. Applications, data storage, and infrastructure are just a few of the remote internet-dependent services CC provides, an internet-based technology. Various technologies, rules, and regulations must be implemented to safeguard the CC technology's software, features, and associated resources. The foundational ideas of CC are pay-as-you-go services for information technology and computing, dynamic scaling, upfront capital and overhead cost reduction, and on-demand computing infrastructure. Popular security methods, particularly symmetric ones, may be widely employed in Cloud Service Providers (CSPs) utilizing encryption techniques due to increased concerns regarding cloud storage and data safeguarding [2]. Scalability, affordability, and manageability are some advantages of CC. Cloud storage also

offers infrastructural features, including ease of use, economics, universality, lease pluralism, dependability, and versatility. A cloud client can request these tools to create, operate, and host services and apps that are adaptable to any location and any kind of device [3].

Cryptography is a process of creating a message that securely alters user data to be conveyed using encrypting and decrypting progress. Information is concealed and securely stored using cryptography to prevent unauthorized access and ensure that only designated individuals may securely share and access it. Security issues are reduced by using encryption, cryptography, and authentication techniques [4-6]. By encrypting data and carrying out only the reverse decryption process, which yields the original text again, cryptography is said to be the art of creating a message that securely modifies the data to be conveyed. CC problems with network data and application security can be resolved with cryptography [7].

Online file-storing models, email, social network spots, and online company applications are all included in the cloud services. Essential features of CC include resource pooling, controlled application, quick adaptability, on-demand self-support, and extensive network connectivity. Clients (individuals or groups) can use on-demand self-service to order and manage their machine services. Access to pooled resources enables the delivery of services over private networks and the internet. Clients using cluster services come from different computer resources, usually located in distant data centres. Quick adaptability guarantees that programs can be modified somewhat. Billing to customers is based on an assessment of product utilization [8]. Although cloud storage has developed into a sophisticated service model, its users' (individuals, enterprises, etc.) access to its technology is limited.

However, there are several issues with cloud infrastructure, mostly pertaining to scalability, multi-tenancy, and interoperability. However, security is the most important concern since cloud infrastructure, like any system that uses networks (e.g., embedded networks, grid computing, etc.), is vulnerable to a wide range of threats [4]. Security issues with CC will prevent it from being widely adopted. The sharing of computing resources in the cloud actually makes it more difficult to maintain their security and prevent unauthorized use or access. This problem primarily disturbs data, i.e., data that is outsourced to the cloud. Among others with CC, the main security concerns are network security involving internal and external assaults and a secure connection between CSPs and their users. Several technologies and protocols are being developed created to guarantee the security of data transfer over networking, with cryptography being the most successful method. Cryptography is the process of converting plaintext into cipher text. It is typically a mechanism for securely sending materials by ensuring that only the intended receiver can locate them [9, 10].

Since just one key needs to be processed quickly, a large volume of data is going to be employed for encryption [11]. There is no established procedure in place for cloud service providers to secure data from intrusions and attacks. Cyber threats target end-user data, which is shielded by the cloud employing cryptographic techniques that make it difficult for an attacker to decipher the cipher text. Compared to small keys, a lengthy key makes it impossible to decode the classified text, making it more secure. Thus, the key generating progress seems to be more scrutinized and needs to be optimized effectively [12-14]. A new area of study in cryptography called HE is being developed [15] to address security concerns, particularly those pertaining to cloud environments. It enables users to employ encryption to safeguard their data on the cloud, allowing the cloud to handle encrypted data; sending the data back to be decrypted is unnecessary because it is maintained encrypted in the cloud. Additionally, HE enables users to distribute encrypted queries through the cloud, which can process encoded queries across encrypted data and return encrypted answers to users. From there, users can decrypt the answers to obtain the necessary result, as demonstrated clearly.

Data centres have historically maintained and controlled the data but lack security. They are vulnerable to several kinds of attacks. As a result, a reliable system for safe data transformation and privacy preservation in servers is required. Through the use of the many services available on the system, a vast quantity of information is shared, and an enormous quantity of information is kept on the cloud server. Many scholars have recently proposed a range of strategies to improve the operation of the cloud environment and fortify the security of various files. Certain systems currently in use employ hybrid approaches, in which encrypting and decrypting progress are dependent on distinct methodologies [16], client-side methodologies, in which security is solely executed on the client side [17], and numerous tactics, including the application of artificial intelligence [18]. Although they can be somewhat helpful, security must also be provided for the data created by cloud devices that must be stored. Small devices have little processing and storage power; hence, they can only deal with light protocols. They collect the necessary data and perform very minimal computational operations. Light processing devices should be able to use a single standard that is both lightweight and strong enough to protect them [19]. Therefore, a lightweight technique is required for encryption when the encrypted cloud storage is where data will be kept, improving security and ensuring secrecy.

1.1. Problem Description and Inspiration

The complexity of the encryption technique, the resources accessible from the device, the application's security requirements, and authentication across edge computing devices are some of the variables that affect how tough encryption is in cloud-based Internet of Things (IoT) devices.

The problems that occur in the cloud of data have been explained in the case studies. In 2019, the Google Cloud data has been exposed. Sensitive files have been viewed publically. It creates an insecurity in the data stored in the cloud. The main issues, among others, are that their limited processing, memory, and battery life could create challenges when applying sophisticated encryption techniques. Due to the high storage of data, memory consumption is also high.

Hence, there should be any compression-related concept to overcome this issue. The LWC algorithms are specifically developed to be computationally effective and need relatively minimal memory and power utilization; they can enable encryption in those devices [20]. In the LWC Algorithm, if the key is not chosen properly, then security issues will occur. One such tactic is to employ symmetric key encrypting techniques with smaller key sizes and lesser computational overhead. Even still, these devices provide limited memory and computing power, which severely restricts the use of encryption; overall, applying the technique of lightweight encryption may assist with ensuring data protection and safe communication in those devices using compression along with optimization.

For encrypting the multimedia statistics created by those devices, a lightweight encryption technique is suggested [21]. The system requirements for encrypting data are quite low. Since the mechanism is symmetric, decryption also uses the same key. The main advantage of using this is that it is more difficult to refute the logic of the plain and cipher messages because of the flexible lengths. It also makes the system more resilient to various attacks and enhances security in tiny peripheral devices. Previous techniques for protecting LWC data from attacks are effective; however, they still have drawbacks with memory capacity, transmission costs and so on. Through gateways, the network devices communicate data to the server. An optimized key produced by the CS2JDK procedure is appended to the data sent by the Key Selection, Generation and Expansion Block (KGEB). Then, to decrease the data size and thus the transmission cost, a new compression-based LWC is applied. A Feistel model and an upgraded neural network transport the compressed data with added security. The suggested system efficiently selects the most secure path for data transmission.

1.2. Main Contribution

Various encryption techniques are used to safeguard and secure the data stored in the cloud. This study proposes a novel lightweight cryptography technique. It is known as an Advanced Lightweight Cryptographic Algorithm (ALCA) and is used to improve CC environment data security. It uses symmetric cryptography to encrypt data. This paper offers a new technique for providing key generation. The compression of transmitting data can be done through an enriched PCA approach. A contemporary encryption system and compression techniques are given in this submitted work. The data gathered from each device is coupled with the security key created utilizing the efficient key (S2JDK) generation procedure to increase network security. Network devices read data, add sensed data, and create cipher text by generating a security key utilizing the efficient key generation process. After that, a compression strategy is performed on this message before it is transmitted. Similarly, the next block combines the message after receiving the compressed cipher text message from several bits. The LWC technique is used when forwarding the accumulated data to provide security. Several analytic architectures in that environment rely on statistical analysis, entropy change analysis, computational time, and key sensitivity. It is developed to address the issue at hand by protecting user privacy in cloud environments. The overall modules involved in the submitted work are provided below.

- In this work, modern compression and integrated encryption schemes are presented.
- The data is categorized into three sensitive forms for quick access.
- To compress the transmitting data through an enriched PCA approach for consuming memory storage.
- To offer a new circular-based technique for providing key generation.
- The observed data is coupled with the optimized security key, i.e., obtained by the most efficient key (CS2JDK) generation procedure to increase network security.
- CS2JDK is used for the key generation process that is especially designed to produce secure cryptographic keys for encryption/decryption operations in the cloud for security purposes.
- To deliver prompt service and superior results in comparison to the pertinent methods.
- To focus on study performance employing various analyzing metrics, including memory, computing time, changing entropy, and statistical analysis.

The general concepts involved in the cryptographic data, their issues, other relevant studies, the newly submitted model, their simulation, their performance evaluations and conclusions are all provided in the upcoming sections.

2. General Concepts

2.1. Cryptography

The process of encrypting data on storage devices using cryptography theory is known as data encryption. The study of mathematical methods of information security, including entity authentication, data integrity, secrecy, and data origin authentication, is known as cryptography [12-14]. Systems for cryptography can be divided into three categories: symmetric methods (also referred to as standard or secret keys), asymmetric methods (sometimes known as public keys) and hybrid. Using the same key by both the sender and the recipient parties is known as symmetric encryption, such as

Data Encryption Standard (DES) and Advanced Encryption Standard (AES). The sender uses this key only once to encrypt data, and the recipient uses it once more to decrypt it. Two keys are required for asymmetric encryption: a public key (for encryption) and a private key (for decryption). The publicly available public key encrypts the data. The private key, which needs to be kept a secret, is then used to decrypt the data like the RSA concept. Employing a mixture of symmetric and Shared data storage and processing resources are made exclusively available through cloud computing, an internet-based computing model.

Processing the gathered data on different cloud providers that have limited computing power has become more and more necessary. The realization of the algorithms for encrypting data in cloud computing must be homomorphic and lightweight, as they cannot handle large amounts of processing. The primary disadvantage is that it requires users and businesses to grant third parties access to their confidential information. A new area of cryptographic study called the LWC framework is designed with the generation of an optimized key using the CS2JDK model.

Controlling the exploration and exploitation search to prevent issues with harmonic convergence or being caught in local optima is its most difficult assignment. Our goal is to enhance the ability to be confined to local optima by introducing a new orthogonal learning-based variation of the search optimizer. Feistel encryption routing involves the transfer of compressed data that has been added by this key. Next, data aggregation based on compression is used to lower the data size and, therefore, the transmission cost. Data size reduction can be done by utilizing the enriched PCA-based compressing scheme before transmitting data. A generated key produced is affixed to the data that the device transmits. The main emphasis of this work is to investigate guaranteed cloud data security and outperform other relevant systems on cloud data using new optimizing techniques.

Asymmetric encryption is used to maximize the strengths of every type; hybrid encryption combines one or more encryption methods. This hybrid cryptography technique was developed to offer a secure and effective encryption mechanism. In cryptology, lightweight block ciphers have become more and more important since the introduction of the DES. Numerous LW architectures and VLSI are made for situations with limited resources. The majority of them employed ECC randomization, cellular automata, evolutionary algorithms, and chaotic maps in AES. LW block ciphers are crucial components in developing numerous cryptographic protocols and are frequently used to offer bulk data encryption. The innovative, efficient LWC method is used in the first layer, and multiplicative homomorphic schemes are utilized in the second layer to enhance secure data in CC. This method provides features that apply to both symmetric and asymmetric cryptography.

2.2. Cloud Computing Security Issues

All approaches aimed at protecting, restoring, and guaranteeing security against diverse attacks are included in computer system frameworks' cloud security of data. Data security issues concerning fundamental security needs for an innovative technology solution that is safe and effective in CC. It significantly improves data security with the help of infrastructure, software, network and storage security. In their world, the reality of outsourcing software and data transfers control away from the user and onto the CSP. The act of having faith and dependence on someone or something to act in a way that fulfils promises is known as trust. Trust is a multifaceted concept in computer science, encompassing aspects like distributed system dependability, accessibility, and security control in computer networks. Trust depends on the cloud vendor and the implementation architecture.

The unique architecture of CC efficiently addresses many conventional security vulnerabilities [28]. Figure 1 illustrates how CC considerably requires data security. However, several unique security concerns have been introduced by its unique infrastructure. The five security supports are confidentiality, integrity, authorization, authentication and availability. These characteristics are now essential when building secure systems, particularly when it comes to CC architecture.

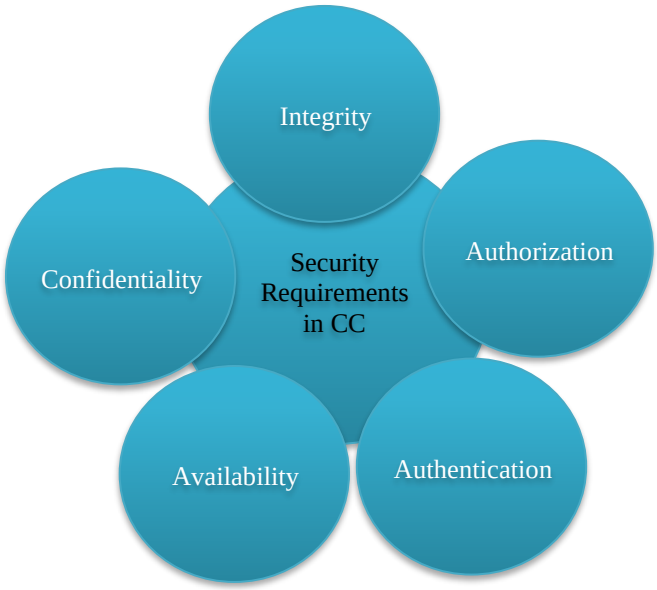


Fig. 1 Security requirements in the cloud computing model

2.2.1. Confidentiality

Only systems or parties with permission to look into protected data are considered to be in possession of confidentiality. Data breaches are more likely when data is outsourced, given supervision to a cloud provider, and made available to many parties. Multi-tenancy, data persistence, application security, and privacy are some difficulties. The cloud feature of resource sharing is known as multi-tenancy.

Numerous challenges to secrecy and privacy arise from the CC architecture, which shares various types of resources to allow several clients to utilize the same resource simultaneously.

2.2.2. Availability

A system's quality allows an authorized entity to access and use it when needed. The ability of a system to function even in the event of misbehavior by some authority is referred to as system availability. The system must be able to function in the event of a security breach in order to guarantee availability. The user in a cloud environment depends on the network's availability as they are relieved of their need for hardware infrastructure.

2.2.3. Integrity

It pertains to the ability of authorized parties to make approved modifications to data, software, and hardware. The term "data integrity" describes safeguarding data from illegal additions, deletions, or fabrications. An assurance to make sure that data is processed and transmitted without being changed or modulated, and requests from customers to add, update, duplicate, or remove data are only accepted.

2.2.4. Authorization

The system uses it to decide what degree of access to secure resources a certain authenticated user deserves. Authorization is crucial to ensuring data integrity in cloud environments because of the increasing number of parties involved.

2.2.5. Authentication

By using insurance on their accounts, customers may ensure their integrity while granting access to details, which is a requirement of authentication. For these reasons, CC has emerged as a promising avenue. Data security is crucial to safeguarding all of the features and advantages CC and the Internet offer. With LWC-based encryption technology and optimization to provide data confidentiality, it can be ensured over the network. The goal of this network is to offer a set of security functions that can guarantee the system's confidentiality.

3. Literature Survey

The use of CC is revolutionizing computer technology. It tackles several common computing issues, such as scheduling redundant computation periods, downloading software updates, and controlling peak demands. CC significantly impacts both the market structure and every area of our lives. Data security holds significant importance in their context. A variety of strategies are employed to ensure secured data. These techniques can be separated into four groups: encryption, backup, erasure, and masking. Cryptography appears to be the best method for securing data that is outsourced in an unstable environment, such as CC, when physical control over the data is lacking. Because of the features of many tenancies and the ease with which providers can access data, it is forced to maintain anonymity using creative methods that mostly rely on access control and encryption. Numerous research projects have been proposed to safeguard cloud data. To provide cloud security and privacy, the author [22] proposed a hybrid cryptography strategy that included symmetric and asymmetric key approaches. Using key creation, and this method will be applied to encryption and decryption processes. Modern Elliptical Curve Cryptography (ECC) has improved cloud data encryption.

Crypto Membranes, a group of native client-side elements that facilitate the development of online applications and serve as a dependable barrier between the client and the server, are introduced [23]. User data is encrypted on the client side and possibly untrustworthy third parties while maintaining total compatibility with current client-side development procedures. Furthermore, to provide a faithful demonstration, he demonstrates how web browsers can use an already available crypto membrane by using a conventional browser throughout the prolongation of the transition time. A pattern that exemplifies the vulnerabilities their design faces and the corresponding countermeasures are put out by Fernandez [24]. The design offers security protection for data assets and communication channels to fend off attacks and streamline security administration.

The security measures include an Intrusion Detection System (IDS), a secure channel, a security logger/auditor, and authentication. This adds to a set of patterns that their team is creating for IoT ecosystems with an emphasis on security. In a hybrid cloud environment, data protection via encryption is critical. A wide range of encrypting procedures has been accessed; however, they also raise data security concerns. The Attribute-Based Encryption Model with Dynamic Attributes Supporting (ABE-DAS) has addressed these concerns [25]. It can work with both organized and unorganized data. It converts plain text to cipher text by utilizing both static and dynamic attributes. There is a significant increase in CC and IoT devices that use cloud systems, which lead to simple, effective, and safe data access. It has been concluded that a lightweight ABE technique is necessary for the cloud-based IoT.

Communication security [26] is created and developed, i.e. satisfied by straightforward techniques. The objectives are to analyze the best LW ciphers and shape the features of their construction. The design needs to pretend to build the cipher within accessories that allow for the monitoring of several metrics. Several ciphers were used in order to accomplish the specified purpose. Field Field-Programmable Gate Arrays (FPGA) and Application-Specific Integrated Circuits (ASIC) technologies were utilized in the modelling, implementing, and optimizing of these ciphers on a platform. Many parameters have been considered, such as block sizes,

implementation rounds, and important scheduling factors. For distributed Wireless Sensor Networks (WSN), a flexible, secure compressive sensing-based data collection technique that offers increased efficiency and security is given. It also lowers the cost of communication. The networked devices recite the data, add the sensed data, and create cipher text by generating an optimized security key [13] utilizing the Spider Web-Based Dynamic Key (SWDK) generation process.

Next, a PCA compressing algorithm is performed on this message. A couple of key variables are used to analyze their effectiveness. Significant factors taken into account are key space, running time, complexity, and correlation [27]. The research [28] presented a New LWC Algorithm for secured data that might be used to protect CC platforms. To lower the complexity of the encryption, the approaches encrypt data using a 128-bit (16-byte) block cipher with the same size key. [29] describes a lightweight encryption technique for encrypting data in network devices. It uses an 80-bit key and a 64-bit block cipher. After analyzing all these techniques, the modern LWC-based technique can be built using an advanced compression model with the optimized key-generating module to overcome the above-revealed issues.

4. Proposed Module

4.1. Need for Jellyfish Search (JS) Optimizer

There is a vast range of scales in ocean currents. These phenomena have allowed the jellyfish species to spread out practically everywhere in the ocean, in addition to the individual movements of each jellyfish inside the swarm and their subsequent migrations along the ocean current to produce jellyfish blooms. The optimal location would be found by comparing food quantities because jellyfish visit different areas with varying amounts of food. Consequently, a new algorithm is devised here that draws inspiration from jellyfish's search behavior and mobility in the ocean. JS optimization is the name of this tool.

The activities and motions of jellyfish in the ocean are theoretically modelled in the spherical subsection for analyzing their directions, after which an optimization method is created according to the mathematical model. Similarly, the input LWC model has the number of bits. There is a need to divide these bits into specific ranges. Also, an integrated Feistel model makes it possible to encrypt and decrypt enormous amounts of data quickly and effectively. Numerous linear and non-linear changes are carried out by it. However, using a predetermined sophisticated technique, this process ensures that the input and output cipher bits have a clear relationship and are dependent on one another. Also, the spherical form of the key-generating model is utilized to cover all the bits, even if there are any linear or non-linear modifications. Thus, the spherical-based JS optimizer is chosen to provide the secured key in the submitted model. Furthermore, the security keys do not match, and the data is inaccessible to unauthorized users or receivers, making it impossible for them to retrieve the information transmitted. Using new compression contributes to reducing data size, which lowers energy and transmission costs. In order to solve the issues mentioned above and create an innovative and reliable multimedia data security paradigm, LWCs have been designed in this article.

4.2. Key Selection, Generation and Expansion Block Model

In this model, the data are scrutinized into three categories: public, medium (data from group users), and highly sensitive data (i.e., owner-only data) with temporal and association rules derived by the rule mining algorithm.

Two distinct algorithms are used for the encryption, with a unique ECC being used to encrypt the most sensitive data. The non-sensitive data are kept in plain text format, while the medium-sensitive data are encrypted using an encryption technique based on the CS2JDK-KGEB. This study proposes a new algorithm for key selection and management that chooses the keys and encryption technique based on the data's sensitivity. The data are encrypted before being stored in a cloud-based database and replicated in several data centres [34]. The most extremely sensitive data are encrypted using ECC with a 100 base table (ECC-B100), and the highly sensitive data are encrypted via the CS2JDK-KGEB-based encryption method, according to the design of the new key management algorithm. As a result, depending on how sensitive an attribute is, this suggested algorithm is utilized to encrypt the data at different security levels.

Using association rules with temporal restrictions, the database is separated into fragments based on the three sensitivity levels of the data. They state that the non-sensitive data is kept in plain text format for public access, the medium-sensitive data is encrypted using the CS2JDK-KGEB algorithm, and the highly sensitive data is protected using the new ECC technique. When the new key is issued for each allowed user based on their allocated key, all accessible data is encrypted using the Base100 table. The key can be used to retrieve the saved data in its encrypted format.

In contrast to traditional cryptosystems, the submitted model aims to prevent attacks on cloud networks by retaining a larger degree of confusion and a lower bit size or key size (in this case, consider 64 bits). The suggested security structure can be viewed with less computational overhead as a hybrid prototype. It takes advantage of the strength of block cipher techniques similar to Substitution and Permutation Networks (SPN), Reformed Wavelet Transform (RWT) and Feistel architecture to implant a higher degree of attack-resiliency using a symmetric block cipher technique. The sensitive data is compressed using an advanced PCA-based, normalized orthogonal model. This learning technique uses limited trials to extract more meaningful information utilizing recent solution vectors and to conduct exhaustive trials to forecast the optimal combination. Three major sequential

processes make up the functional structure of our suggested model: key creation, encryption and decryption. To ensure that the SPN block cipher satisfies the required Shannon for both confusion and diffusion, the first iterative-alternating rounds of SP similar to transposition can be conducted. To accomplish this, an optimizer, namely CS2JDK-based KGEB, has been created to guarantee that any alterations to the encryption are pseudo-random. KGEB has been used across five rounds to increase the degree of ambiguity and the security framework. The goal of each round is to satisfy Shannon's confusing and diffusing circumstances specified above.

The suggested model guarantees greater confusion and less processing in this way. The suggested paradigm makes it possible to modify the cipher text in a way that prevents the original data from being easily exposed to an intruder. Integrated Feistel topology is used with SPN to encrypt and decrypt the input data to increase performance even more.

Notably, using Feistel and RWT architecture [36] minimizes extra computational strain by enabling decryption in the same manner as encryption. In contrast to existing ciphers like DES, Blowfish, etc., the suggested feistel architecture allows quicker and more effective encryption decryption over enormous data sizes. Since this architecture is based on iterative-alternating rounds, the new CS2JDK is employed with a time control module. The proposed security model can be described as a hybrid cryptosystem that operates using a 64-bit key, and plain text is used in a symmetric key block cipher. Our suggested encryption mode operates on a 5-round iteration mechanism that generates a diffusion matrix and confusion by operating over predefined mathematical functions (KGEB).

Higher rounds ensure a higher level of security but at the expense of enlarged computational overheads. Conventional cryptographic methods typically apply an average of 20 rounds to encrypt objective data, which can result in a significant amount of computational overhead and time exhaustion.Since every node in a CC system functions as both a key generator and a decoder, it's critical to keep computations as low as possible.

A mathematical model is created to do this by concatenating the data (using ++ or ||) and applying the logical functions XOR ($\oplus$) and XNOR ($\odot$). Key generation is the most crucial part of the key-generating and expanding process since it is the primary responsibility during encryption and decryption.

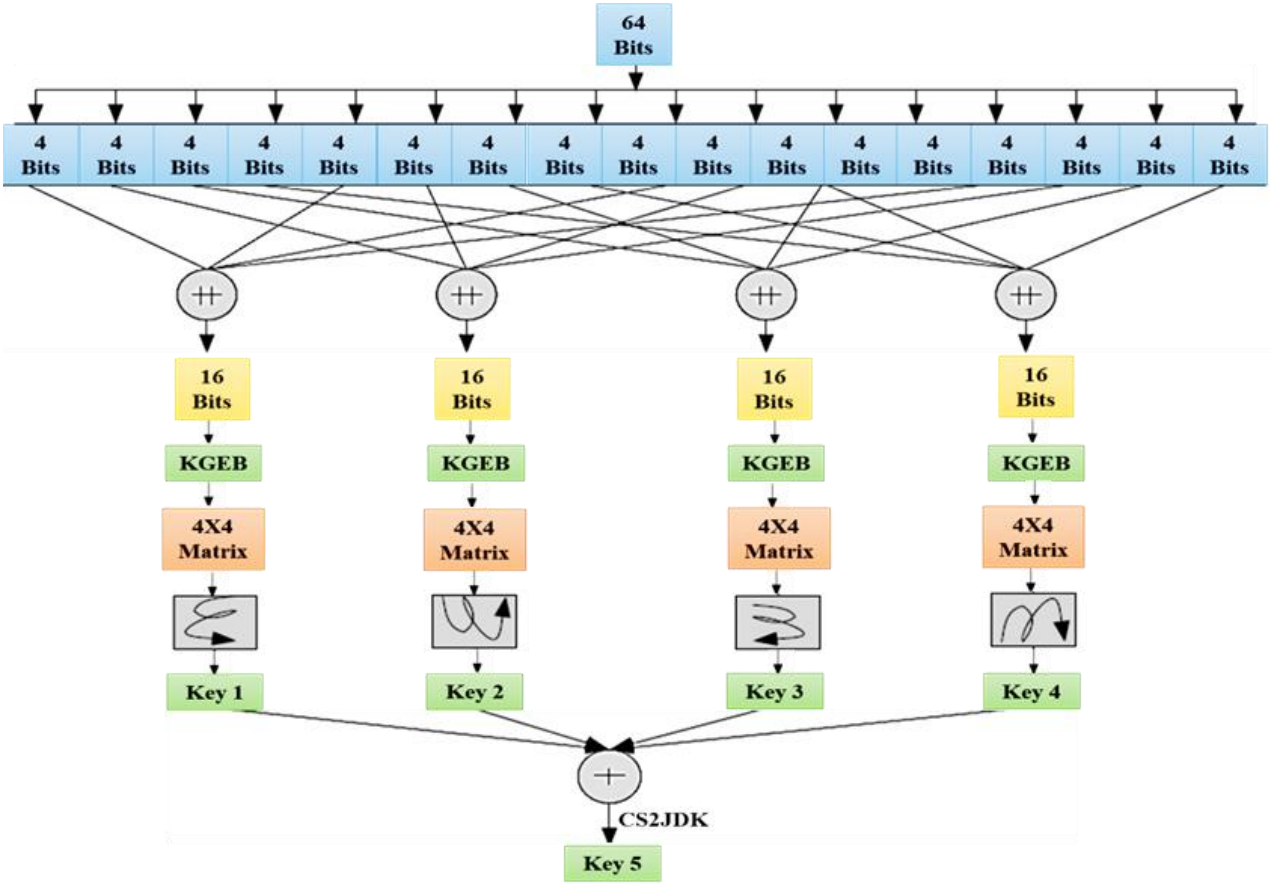


Fig. 2 CS2JDK-based KGEB framework

Feistel block (FB) architecture-dependent encryption is used and executed in several rounds (as 5), requiring a different key for each round (5 distinct keys) in order to accommodate it. In order to do so, a function known as the "CS2JDK-based KGEB framework" can be created. The dual-way encryption is done in FB. They can be split up into two ways, odd and even forms, and then the encrypted output is provided to the RWT block (RWTB). This block decomposes the image into four band parts: low-low, low-high, high-low, and high-high bands. The high band combinations are ready to hide in integrated FB. The resulting stage image becomes accessible to an insecure channel after the encrypted text has been incorporated into the cover image. To recover the confidential data, all procedures are carried out in reverse order at the recipient's end. Finally, the inverse transformation process is performed to obtain the image.

Our new secured prototype is built as a 64-bit block cipher, which implies that to accomplish the encryption of the 64-bit input data, a same-bit key is required. This encrypted key, K_c, is applied in our suggested model. It was then utilized as the initial value of the KGEB block, which performs mathematical operations to cause confusion and diffusion before retrieving 5 different keys. As a result, it generated 5 unique keys for every encryption cycle, which are then used in the crypt procedures. Initially, the 64-bit input data is divided into 16 separate chunks, each consisting of 4 bits, as shown in Figure 2. The next four unique 4-bit chunks are concatenated to form a 16-bit data chunk, for which the Feistel network acquires the keys. Following KGEB execution across the 16 bits of the merged input, a matrix with 4 bits per row is produced. The following steps make up the entire KGEB process:

1. Divide the user-provided 64-bit input cipher (K_c) into four separate 4-bit segments.
2. Notably, 16-bit data can be obtained for every block upon which KGEB is applied, with 4-bit separate segments. After processing KGEB, 16 bits of data is acquired. Then, equation (1) is used to carry out the first substitution of the segments or blocks of K_c. Similarly, 16 bits of data are obtained for each KGEB Function.

$$Kp_{n\in1,2,3,4,}f = ||_{m=1}^{4} Kq_{4(m-1)+n} \quad (1)$$

In equation 1, the variable n =1 to 4 for the initial 4 rounded keys.

3. After collecting the values for $Kp_{n\in1,2,3,4,}f$, it underwent additional processing to produce Kg_if for every 16-bit block. Ka_if is obtained mathematically by using equation (2).

$$Ka_if = f(Kp_nf) \quad (2)$$

As seen in Figure 3, the KGEB function is created by strategically combining the dual functions LF and NL. The functional schematic depicted in Figure 3 presents a Non-Linear Function (NL) and a Linear Function (LF) in that order. The transformation mechanisms for both NL and LF are provided in a manner similar to the previously mentioned systems [28-30].

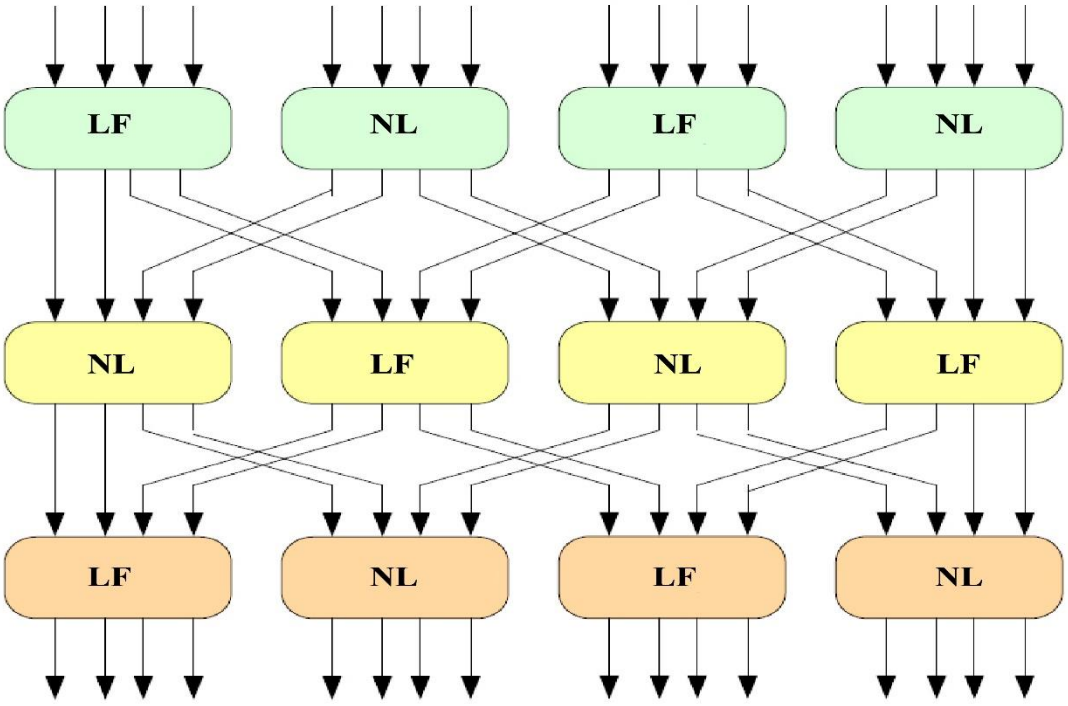


Fig. 3 KGEB function

4. Confusion and diffusion transformations, both linear and non-linear, made of LF and NL, are observed in Figure 3.
   - Each *f*-function in the output is produced from 16 bits in a single matrix, 4∗4. The result is a key matrix with four arrays designated M_{k1}, M_{k2}, M_{k3}, and M_{k4}.
   - The circular keys are represented by the arrays (K_1, K_2, K_3, K_4).
   - The arrays are rotated to produce KM_1, KM_2, KM_3, and KM_4 as the result. Each created key combination results in the four public keys (Kk_1, Kk_2, Kk_3, Kk_4).
   - XOR is applied to the four existing round keys to get the fifth round key.

By getting the values of the LF and NL functional blocks, encryption has been carried out by applying KGEB in the same manner as indicated in Figure 2. The output of the KGEB Function is re-sampled in a 4∗4 matrix once the KGEB Function associated with each 16-bit block is obtained.

5. The key matrix has been divided into four separate arrays of 16 bits to subsequently retrieve the keys for each of the 16-bit blocks (KM_1, KM_2, KM_3, and KM_4). These arrays are referred to as "per-round-key (PRK)" here. Equations (7) through (5) show the four different keys and the ordering of associated attributes. Notably, the concatenation is shown by the operator # in the equations below.

K_1 = p4#p3#p2#p1#p5#p6#p7#p8#p12#p11# p10#p9#p13#p14#p15#p16 (3)

K_2 = q1#q5#q9#q13#q14#q10#q6#q2#q3#q7#q11# q15#q16#q12#q8#q4 (4)

K_3 = r1#r2#r32#r4#r8#r7#r6#r5#r9#r10#r11#r12# r16#r15#r14#r13 (5)

K_4 = s13#s9#s52#s1#s2#s6#s10#s14#s15#s11#s7# s3#s4#s8#s12#s16 (6)

Using the four unique keys (each for one round), the above equations 3-6 are obtained. The security keys are extracted from these four created keys at the spots where the single key (K_5) connects. The numerical values of the intersection points are padded to create the keys' frame. The data are attached with the security key created by the CS2JDK unit and then communicated by the device.

The compression method is used to lower the size and transmission cost of the data because the totalling of a security key grows its size. The creation of random numbers is not addressed by the suggested security plan. As a result, the attacker finds it difficult to get the key. The key values have now estimated and executed an XOR logical operation to create a fused key (key 5) in equation 7.

$$K_5 = \bigoplus_{n=1}^{4} K_n \quad (7)$$

This key is optimized using the CS2JDK model, which will be explained in the upcoming section.

4.3. Jellyfish Search Optimizer (JSO)

Around the world, jellyfish can be found in both shallow and deep water [31, 35]. Their bell-shaped, soft bodies are covered in long, stinging tentacles on the underside, which they utilize to either sting and paralyze their prey (small fish or plankton) or catch them. They come in an extensive range of hues, dimensions, and forms. Every one of the numerous species shows unique adaptations to the marine environment. Jellyfish possess characteristics that enable them to regulate their motion. They push water out of their bodies by contracting them like an umbrella, which helps them move forward. Even with this capability, they mostly rely on tides and currents to move in the water.

When the right circumstances arise, jellyfish can swarm and form a massive mass known as a jellyfish bloom. The formation of a swarm is influenced by several variables, including temperature, oxygen availability, accessible nutrients, predators, and ocean currents. Ocean currents are considered the most significant of these factors in forming a swarm. The way jellyfish search and move in the ocean served as the model for the jellyfish search algorithm. Three potential courses of action during the jellyfish movement are listed below.

- One of the two ways jellyfish move is either with the ocean's current or as a group; the transition between these two ways of moving is controlled by a "time control mechanism."
- In the water, jellyfish migrate in quest of food. They are more drawn to that location where there is more food available.
- The location and related objective function dictate how much food is found there.

4.4. Ocean Current

The jellyfish are drawn to the nutrient-rich ocean current. Equation 8 describes the trend or direction of the ocean current.

$$\overrightarrow{Jtrend} = X_j^* - \beta * rand(0,1) * \mu \quad (8)$$

Where X_j^* is the current best jellyfish location in the swarm, μ is the mean value of previous jellyfish locations, and β >0 is a distribution coefficient relevant to its length. Concerning the experimental outcomes [31], β = 3 is attained, and their updated positions are provided by:

$$X_{ji}(t+1) = X_{ji}(t) + rand\ (0,1) * \overrightarrow{Jtrend} \quad (9)$$

4.5. Spherical Scyphozoan Jellyfish (S2J) Swarm

A swarm can be a big group of jellyfish that go in different directions, either in passive (type A) motion or in active (type B) motion as follows:

1. Type 'A' motion refers to jellyfish locomotion in and nearby their own places. The latest location of every jellyfish is offered by:

$$X_{ji}(t+1) = X_{ji}(t) + \gamma * rand\ (0,1) * (U_b - L_b) \quad (10)$$

It is feasible to compute the locomotion factor, gamma, and the length of motion with respect to the jellyfish, γ=0.1. U_b and L_b stand for the search space in upper and lower bounds, correspondingly.

2. To replicate the type B action, a jellyfish (j) that doesn't catch our attention is chosen randomly (Equation 11). To determine the direction of motion, a vector is selected from the jellyfish of interest (i) to the chosen jellyfish (j) (Equation 12). When there is more food at the chosen jellyfish (j) location than there is at the other site, the interested jellyfish (i) travels to the first place. On the other hand, the selected jellyfish (j) will quickly leave that location if it has less food than the jellyfish of interest (i). Equation 13 is used to update and report the precise location of a jellyfish.

$\overrightarrow{step}$ is simulated as rand (0,1) of $\overrightarrow{direction}$ (11)

$$\overrightarrow{direction} = \begin{cases} X_j(t) - X_i(t) * (\sin\alpha + \cos\alpha), & if\ ff(X_i) \geq ff(X_j) \\ X_i(t) - X_j(t) * (\sin\alpha + \cos\alpha), & if\ ff(X_i) < ff(X_j) \end{cases} \quad (12)$$

Where ff is the fitness function of position X_i, α denotes the direction in which the spherical process of searching ought to move. In the search space this optimizer provides, the improved search procedure offers deeper investigation and faster convergence. The circular dynamic search modifies the manner in which many searches are carried out by introducing a shift in the search process.

$$X_i(t+1) = \overrightarrow{step} - X_i(t) \quad (13)$$

4.6. Time Controlling Scheme

The ocean current, which frequently comprises enormous nutrient-rich plants, attracts jellyfish. The jellyfish belonging to the swarm migrate into a different ocean current as the meteorological variables (temperature, wind, etc.) alter the ocean current and other jellyfish swarm forms.

Within a swarm of jellyfish, type A and B movements are alternated. Type A is preferred initially, while type B gains preference with time. Equation (14) with time control mechanism governs this shift by controlling the jellyfish's movement both within the swarm and in the ocean current.

$$Jc(t) = |(1 - \frac{1}{max_{iter}} * (2 * rand(0,1) - 1))| \quad (14)$$

Where max_{iter} is the maximum iterations or populations as an initialized parameter, and t is the number of iterations indicated as a time.

The function that simulates moving inside a swarm, whether passive or active, is $(1-J(t))$; X moves passively when $rand\ (0,1) > (1 - J_c(t))$. If not, then X can move actively. The complete model is provided in Algorithm 1.

Algorithm 1: CS2JDK Model

1. Determine initial parameters.
2. Construct population X_i, (i=1,2,…,N) utilizing a logistic chaotic map.
3. N is considered to be the maximum generation of dynamic keys.
4. Compute fitness function (ff) for Xi.
5. Set t=1
6. repeat
7. for i=1 =1 to N do
8. Update J_c (t) according to Equation (14).
9. if J_c $(t) \geq 0.5$ then
   (1) Update active motion.
   (2) Update directions.
10. else:
11. if $rand > (1 - J_c(t))$ then
   (1) Update according to Equation (10).
12. else:
   (1) Update the direction of Xi according to Equation (12).
   (2) Update Xi according to Equation (13).
13. Check boundary conditions for Xi.
14. Compute ff for Xi .
15. t=t+1
16. until$t > tmax$
17. Allocate the best solution X_{fb} =X and ff(X_{fb})=ff(X).
18. Allocate t=1 as initial value.
19. while$t < tmax$ do
20. Discover the neighbour key bit Y for the X.
21. Compute ff (Y) for Y.
22. if ff (Y)< ff (Xi), then
   (1) Xi=Y.
23. else
   (1) Update Δ=ff(X_i)−ff(Y).
   (2) if ($p \leq k_4$) then // total no. of keys
   (3) X_i=Y.
24. if ff(X_{fb})>ff(X) then
   (1) X_{fb}=X;
25. Set t=t+1.
26. Output: X_{fb} // best key generation

First, a random solution with a starting value of X_i is generated, and a new solution, Y, is found from its neighbourhood bit from the key. The fitness value for both X_i and Y is found. If $(Y) \leq Fit(X_i)$, then $X_i = Y$.

However, SA can substitute Y for X_i even if Y's fitness is not greater than X's. In this step, the stop criteria are examined; if they are not satisfied, the updating step is repeated. If not, the optimal solution discovered thus far (X_{fb}) is given back as per the steps in Algorithm 1. The flow diagram is also provided in Figure 4.

4.7. Constrained S2J Concept

The goal of the constrained S2J dynamic key design (CS2JDK) is to explore the effect of multiple matrix combinational factors on the outcome by identifying the optimal candidate solution by evaluating only a small number of sample matrix combinations. This technique has a concept, i.e., to make use of the fractional qualities to figure out the ideal level sequence.

The ability can be employed to estimate the optimal solution for a system with K factors, each of which can occupy one of the Q bit levels. There are a total of Kq exhaustive experiments to be applied. As a result, the total number of combinations is highly dependent on the total number of key K factors and Q bit levels.

When K and Q are big, it seems difficult to investigate every possible matrix combination. For that reason, because it provides a range of orthogonal arrays with different numbers of levels and important factors, the orthogonal strategy is regarded as an effective constrained S2J tool. Two concepts form the basis of the theoretical CS2JDK design.

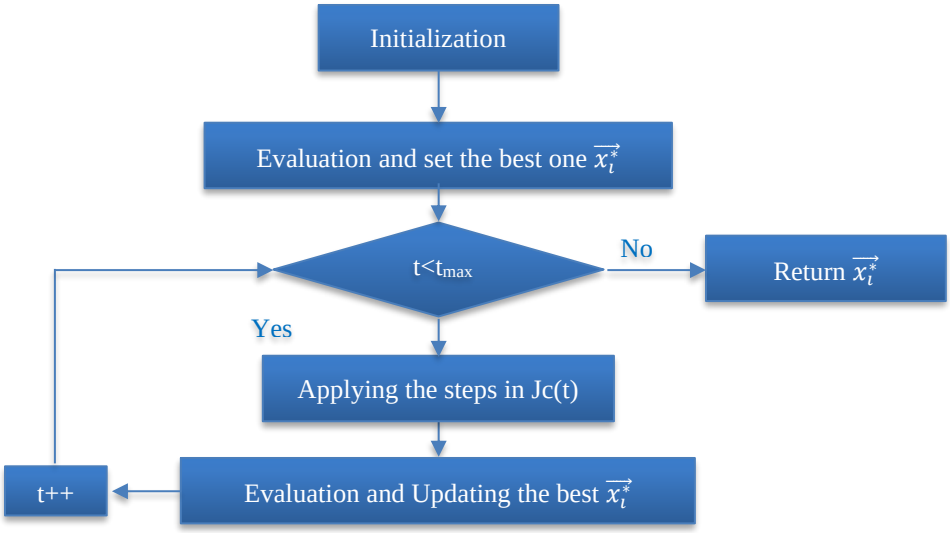


Fig. 4(a) Complete flow of CS2JDK model

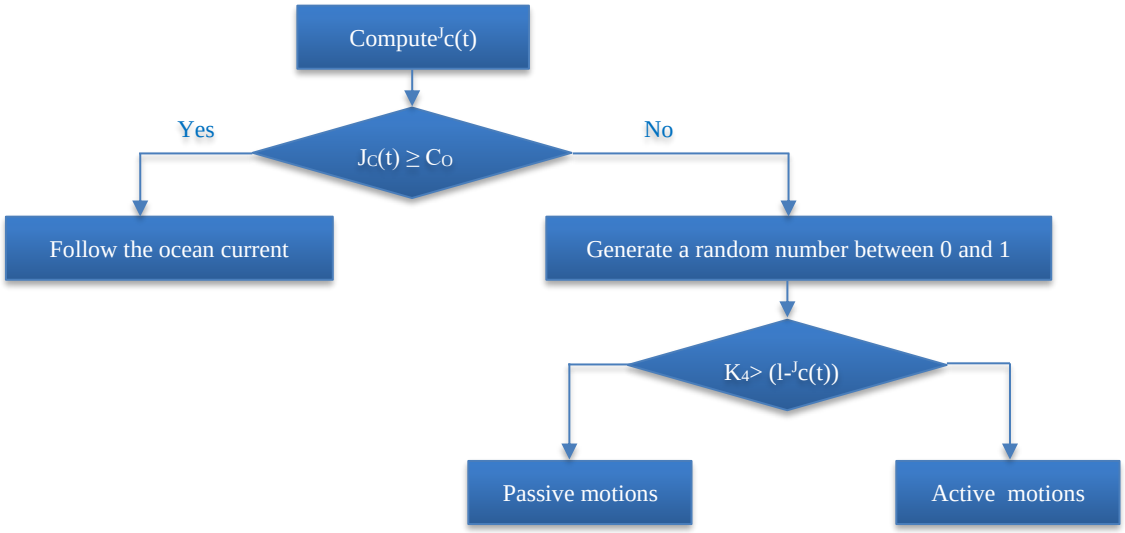


Fig. 4(b) Time controlling concept in CS2JDK concept

- It is written as $C_N(K^Q)$, where Q is the number of levels for each of the key K components, and C and N stand for an array and the overall amount of matrix combinations, respectively, from equations (3-6).
- After all *n* combinations well-known from the constrained orthogonal array have been generated, the optimal combination of levels can be found using matrix factor analysis (MFA). The impact must be ascertained at every key level for every dynamic key element in order to do this. The mean can be calculated as follows.

$$C_{kq} = \frac{\sum_{n=1}^{4} ff(X) \times KM_n}{\sum_{n=1}^{4} KM_n} \quad (15)$$

Where C_{kq} is equal to one if the level of the k[th] factor (k= A, B, C, D) in the n[th] (n = 1 to 4) is q (q=1,2,3) combination and equal to zero otherwise. *ff*(n) serves as the best fitness factor derived result of the n[th] combination of X_i. These parameters are considered from the previously evaluated formulae (section 5.1 and Algorithm 1). Using the four unique keys (each for one round), the above equations (3-6) are obtained. The fused key acts as an aggregator, which is optimized here by considering CS2JDK as,

$$K_5 = \bigoplus_{n=1}^{4} K_n . C_{kq} \quad (16)$$

A new plan is utilized to improve the outdated search system. This approach to resolving the exploration/exploitation issue is predicated on the previous idea. The solutions produced are thought to be the outcome of the novel search technique, depending on the constrained layout. The goal of this process is to increase the common JSO exploitation capability.

4.8. Enriched PCA (EPCA)

The sensor node receives the CS2JDK key $K_n(K_5)$ that the aggregator generates. The attacker node receives the sent key $K_n(K_5)$, but is unable to retrieve it since it is unaware of the KGEB unit with user input parameters.

$$Xj'_n = Xj_n \oplus K_n \quad (17)$$

The advantages of the two appealing characteristics of the EPCA, (i) computational efficiency and (ii) quick convergence when the initialization is near the subspace of interest to construct a low complexity compression strategy [32], are considered. In order to do this, performing an orthogonal iteration is proposed for each incoming subsequence XS_i, with the starting input being the subspace $XE_i{-}1$, which corresponds to the subsequence $XS_i{-}1$.

To be more precise, the symmetric data autocorrelation matrix KA_{ji} for Min Eq. (18) can be substituted when the data block K_{ji} is obtained. This allows us to estimate the subspace of interest XE_i equals the orthogonal normalization of (KA_{ji} XE_i −1). The selection of KA_{ji} will determine whether other subspace tracking techniques are available. The block prediction of the autocorrelation matrix is the easiest and most effective selection.

$$KA_{ji} = K_{ji} K_{ji}^{T} + \delta \cdot I_{ni} \quad (18)$$

Where I_{ni} is the CS2JDK size-n identity matrix (I), and δ is a little scalar number, i.e. utilized to make sure A_{ji} is non-negative definite. The new enriched PCA compressing function compresses the encrypted text data.

$$CXj'_n = f_{cj}(Xj'_n) = \left[\frac{Xj'_n}{cj}\right] \quad (19)$$

Following CS2JDK, the compressed (CXj_n) form of the encrypted data, Xj'_n is mapped from a space of high dimensional into a lower dimension. The compression factor, denoted as cj in this case, is calculated as the product of the lengths of Xj'_n and CXj_n . Similarly, the decompression function can be used to retrieve the decompressed data in the form of ($\widehat{Xj'_n}$)ˆ.

$$\widehat{Xj'_n} = f_{cj}^{-1}(CXj'_n) = cj \times CXj'_n - \left|\frac{cj}{2}\right| \quad (20)$$

The function provided in equation 20 is used at the KGEB to perform the decompression operation.

4.9. Key Retrieval

Each KGEB round keys are retrieved, when data encryption is carried out. Various logical operations are used, including entity-swapping, replacement, and shifting (left-right), to reduce confusion and dispersion throughout the encryption process.

The 64-bit plain text (*Pt*) is divided into four different chunks of 16 bits each for the first round of operation (P*x*0–15, P*x*16–31, P*x*32–47, and P*x*48–63). As each round through the bits operation, the suggested model swaps bits to lessen the originality of the data by changing the bit order. It caused more ambiguity in the encrypted text.

In the meantime, it carries out a bitwise XNOR logical operation among P*x*0–15 and the appropriate round key K_i. K_i and P*x*48-63 go through this process again to produce Ro_{11} and Ro_{14}, correspondingly. After the output from the XNOR logical operator is obtained, it is passed to the KGEB block, which produces the two separate outputs, Ef_{l1} and Ef_{r1}.

Noticeably, the KEBF used for encryption is identical to the KGEB (key expansion function), including substitution followed by swapping. The bitwise XOR is now used to generate Ro_{12} using the obtained Ef_{l1} & P*x*32-47, and the same among Ef_{r1} and P*x*16-31 provided Ro_{13}. The earlier procedures are meant to make the coding more complex. Eq. (21) repeats the identical methods for each round.

$$Ro_{i,j} = \begin{cases} Px_{i,j} \odot K_i; j = 1,4 \\ Px_{i,j+1} \oplus Ef_{li};\ j = 2 \\ Px_{i,j-1} \oplus Ef_{ri};\ j = 3 \end{cases} \quad (21)$$

The final stage of cipher text (*Ct*), which will be used for future communication, is formed by concatenating the final round's results, which is repeated for all subsequent rounds in accordance with (21).

$$Ct = R_{51}\#R_{52}\#R_{53}\#R_{54} \quad (22)$$

The next section thoroughly analyses the simulation results and their implications. Notably, even though the structural sequence reverses the previous (i.e., encrypting progress), our proposed (CS2JDK-based KGEB cipher) model with Feistel structure demonstrates decrypting progress in a similar way and inverse of encryption. It becomes lighter and more economical as a result of the large reduction in total computation.

5. Results Analysis

Utilizing the MATLAB 2023a tool, the suggested crypto-model was created and evaluated on various biological and generic multimedia datasets, including MRI, glaucoma, and diabetic retinopathy. By simulating across benchmark datasets, the suggested security model's performance was investigated both quantitatively and empirically. With the help of an Intel i3 processor and 8 GB of RAM, the entire suggested model was simulated using Windows 2010 as the Microsoft operating system. However, with faster or more sophisticated CPUs like the Intel i5 or i8, the time taken for computation may be longer. The following parts provide a thorough explanation of the performance evaluation of the submitted parts LWC-JDK LWC-CS2JDK with the previous approach LWC [1].

5.1. Statistical Assessment

Initially, looking at the suggested multimedia data security model's statistical performance metrics includes cipher creation, entropy, correlation, and histogram results, in order to evaluate its effectiveness. In order to conduct this evaluation, the performance is examined using several input photos that were representative of everyday life. The standard photos, such as Baboon, Panda and Lena, the widely recognized benchmark data for image analysis, are tested as typical (everyday) life photographs.

Notably, input photos were first transformed from conventional RGB data to gray scale before our suggested crypto-model could run. The input photographs were initially pre-processed, meaning they were shrunk to 256x256 pixels and changed from RGB to gray scale. Notably, the regarded images are in the *.PNG and *.JPG file formats, even though the suggested technique might process any type of input image data.

Following pre-processing, the data underwent the corresponding encryption and decryption processes based on the theory that communications would occur over an erratic cloud framework or platform. As a result, performance outcomes can be generated in terms of entropy, correlation, Unified Average Channel Intensity (UACI) and Number of Changing Pixel Rates (NPCR) using the simulated results and the corresponding histogram outputs. A brief overview of the many statistical and visual assessment parameters employed in the submitted study is provided below before discussing the simulation outcomes.

5.1.1. Image Histogram

Through the use of histogram variation analysis, one can examine the visual impact of a cipher used to encrypt an image and evaluate the degree of unpredictability that remains in the original image following encryption. An image histogram analysis approach can be used to see the added unpredictability in the original image. Here, the histogram difference is kept shortly after encryption to a low or negligible in order to preserve superior security and attack likelihood based on visual perception.

5.1.2. Image Correlation

As a statistical measure, correlation denotes the relationships, dependencies, or correlations between two different variables. A data point or element with a significant reliance typically indicates a significant association. The cipher information must no longer depend on the original image to maintain multimedia security.

The data security aspect is strengthened because no meaningful information can be recovered with such little reliance. In this work, to obtain the correlation coefficient (ϑ) between the encrypted and original multimedia data, eq. (21) can be used. It is necessary to maintain *cov* at or close to zero in order to achieve the ideal state.

$$\vartheta_{p,q} = \frac{cov(p,q)}{\sqrt{D(p)\sqrt{D(q)}}} \quad (21)$$

In which the covariance value is expressed as $cov(p,q)$ and the variances for the variables p and q are expressed as *D(p)* and *D(y)*, respectively.

5.1.3. Image Entropy

The development of ciphers during the security process might greatly increase the disruptions throughout the image input. This can lead to an increase in picture entropy, which can lower the quality of the image while also making it easier for hackers to target particular data. However, to make it more difficult for an intruder to discern between the encrypted and original data, encryption adds extra information to the data. In these situations, preserving the ideal entropy while transmitting the data is imperative. For this reason, the entropy

of each encrypted piece of data is calculated to maintain quality-focused multimedia data protection.

$$\text{Ent}(ł) = -\sum_{n=1}^{2^8} ₱(ł_i)\log_b)₱(ł_i) \quad (22)$$

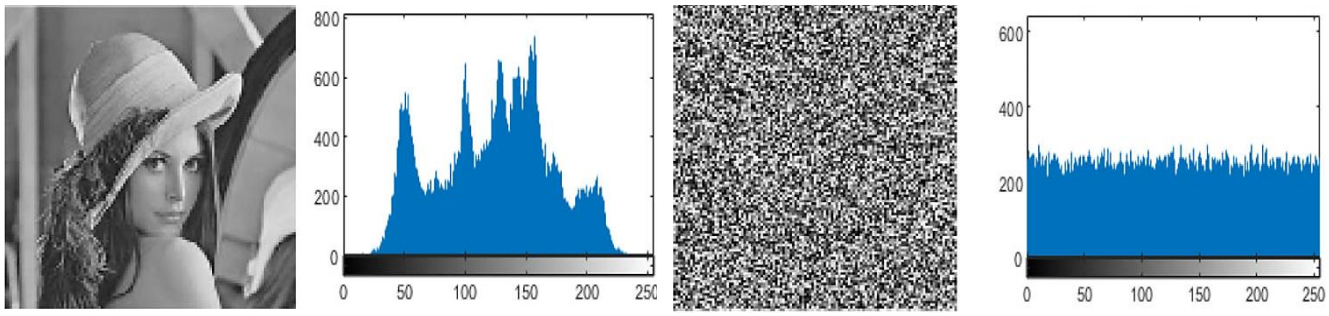
An image's entropy is expressed as $\text{Ent}(ł)$, where $ł$ denotes intensity and $₱(ł_i)$ is the likelihood that the intensity value of $ł_i$ will occur.

5.1.4. UACI and NPCR

The randomness test metrics that are frequently used to evaluate an image encryption technique's resistance to differential attacks are UACI and NPCR. Increased NPCR indicates increased resistance to differential attacks and analyzes more information about these randomization tests [33]. Other types of images can also be processed, such as medical images.

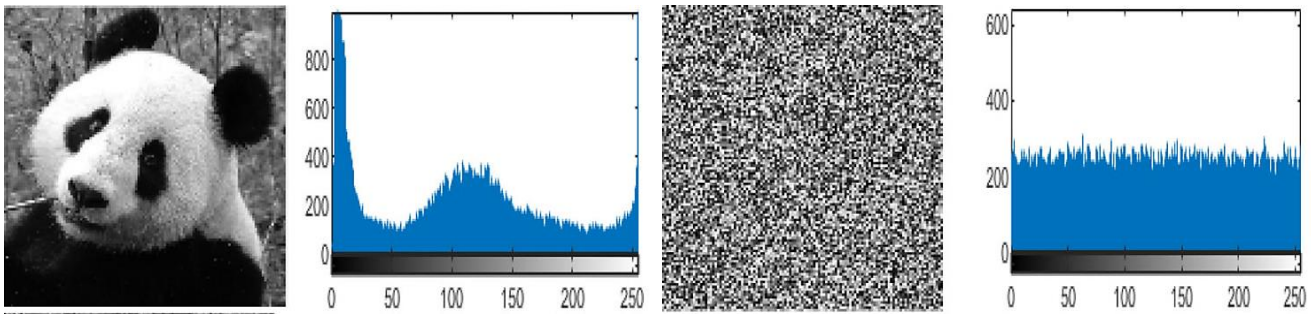
5.1.5. Visual Outputs

Figures 5 to 7 show the simulation findings for the various input data, such as Lena, Panda, and Baboon, using image histograms as the visual outputs. It shows the original inputs, the original histogram, their encrypted image, and the outcomes of the encrypted histogram resultant images correspondingly, as illustrated in Figure 5 to Figure 7 (a-d). The results show that the histogram changes in cipher data significantly after encryption, making it impossible for an attacker to detect when the cipher is unlocked. The entropy level of the original data and the associated encrypted image are shown in Table 1. The entropy level rises for encrypted photos, as shown by these results, which introduces confusion and prevents easy detection of the original data. The findings support the need for an appropriate entropy requirement to prevent attacker discovery in the event of erratic cloud circumstances.



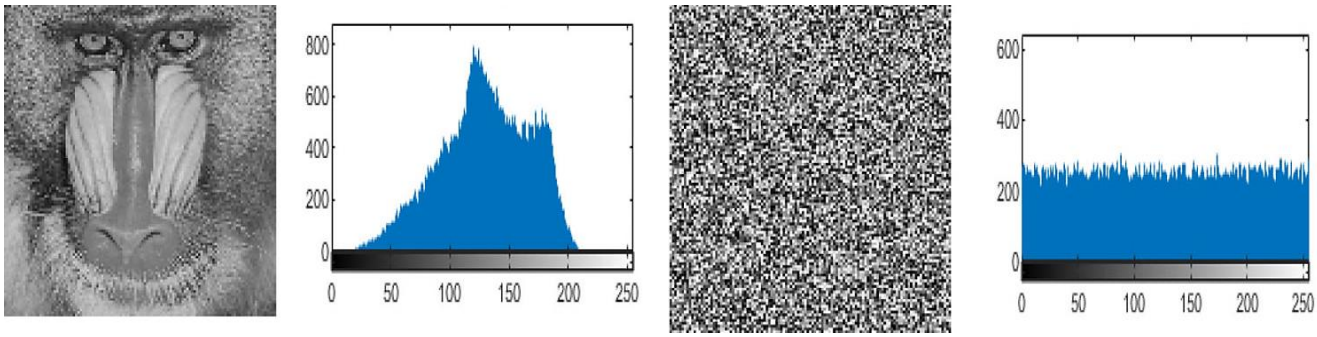
(a) (b) (c) (d)

Fig. 5(a) Original lena image, (b) Histogram, (c) Encrypted image, and (d) Encrypted histogram resultant image.



(a) (b) (c) (d)

Fig. 6(a) Original panda image, (b) Histogram, (c) Encrypted image, and (d) Encrypted histogram resultant image.



(a) (b) (c) (d)

Fig. 7(a) Original baboon image, (b) Histogram, (c) Encrypted image, and (d) Encrypted histogram resultant image

Table 1. Entropy examination

Various Datasets	Input Image Entropy	Encrypted Output Image Entropy		
		New LWC	LWC-JDK	CS2JDK
Lena	7.3904	7.673	7.689	7.7148
Panda	7.4938	7.9968	7.98	7.998
Baboon	7.102	7.8993	7.8998	7.918

Table 2. Analysis of correlation post-encryption

Dataset	Original Image Correlation	New LWC	LWC-JDK	CS2JDK
Lena	0.9763	0.0043	0.004	0.0038
Panda	0.9819	0.0013	0.0012	0.0009
Baboon	0.8199	0.0041	0.0039	0.0036

Table 3. Randomness test analysis

Dataset	NPCR (%)	UACI (%)	PSNR
Lena	99.55	14.67	69.55
Panda	99.59	13.5	59.79
Baboon	99.63	22.8	58.63

For encrypted data, the entropy level rises, contributing to the confusion that prevents easy identification of the original data. Notably, several methods support keeping either high entropy, which is frequently taken into account in steganography-based techniques, or low insignificant entropy.

The findings support the need for an appropriate entropy requirement to prevent attacker discovery in the event of erratic cloud circumstances. Consideration of 8-bit grey scale images in the model can be offered, and this shall have 8 bits as the maximal entropy.

Table 1 shows that, for encrypted images, the maximum entropy attained over three test cases reaches 7.998. The optimum outcome of a successful encryption strategy will be encrypted data with zero overlap. Their comparisons are provided in Table 2. Apart from the previously mentioned visual and statistical characterization, the effectiveness of the suggested security model has also been assessed using the NPCR and the UACI, as shown in Table 3. Higher unpredictability and, hence, greater resilience, contrary to any differential assault probability, are typically indicated by high values of both. The randomization test results that show a higher NPCR value reinforce the proposed system's ability to withstand attacks.

In a similar vein, UACI also attests to acceptable performance. The ratio of the encrypted image to the original image is called the PSNR (peak signal-to-noise ratio). The PSNR is expressed in dB. The encrypted image is more similar to the original than not, the higher the PSNR. Generally speaking, a higher PSNR number should indicate a better-quality image. A good encryption scheme should have the lowest possible PSNR.

5.2. Comparative Assessments

Execution time is one of the most important factors to consider when developing cryptography. The encryption cryptographic execution time is the total amount of time required to encrypt and decrypt the distinct data. Table 4 displays the typical time needed to process the data's encryption and decryption.

Table 4 shows the milliseconds needed to execute comparable algorithms with different file sizes. The suggested algorithm clearly runs faster than the other algorithms, as shown in Figure 8. The efficacy of the submitted algorithm can be assessed using the throughput rate, as shown in Figure 9.

Their performance and throughput are directly correlated; the better the performance, the higher the throughput. Also, the following outcomes, like Structural Similarity Index (SSIM), PSNR, and Mean Square Error (MSE), can be used to determine the encoder technology transmission rate.

The respective values of SSIM, PSNR, and MSE are tabulated in Table 5 and also depicted in Figures 10-12. The maximum better outcomes observed across testing scenarios (for encrypted images) satisfy the cryptographic requirement mentioned above for many applications like secure and quality-focused multimedia data transmission. Therefore, the robustness validates that the suggested security model is appropriate for any type of real-time multimedia communication, including cloud communication.

Table 4. Comparisons of processing time and throughput

Plaintext Size (KB)	Enc/Dec Time (S)			Throughput (kb/Sec)		
	New LWC	LWC-JDK	CS2JDK	New LWC	LWC-JDK	CS2JDK
255 KB	0.67	0.61	0.59	380.94	382.45	391.54
500 KB	0.93	0.903	0.891	537.63	538.69	544.31
750 KB	1.13	1.119	1.102	842	846.7	856.5
1 MB	2.88	2.64	2.58	663.71	667.814	670.4
Average	1.4	1.36	1.214	606	608.8	616.4

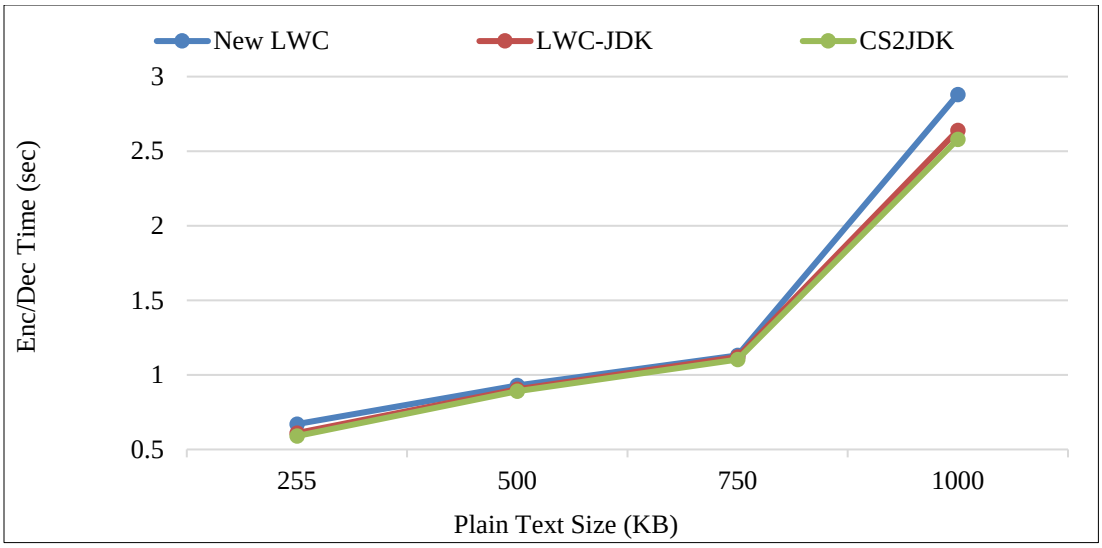


Fig. 8 Comparisons of processing time for various methods

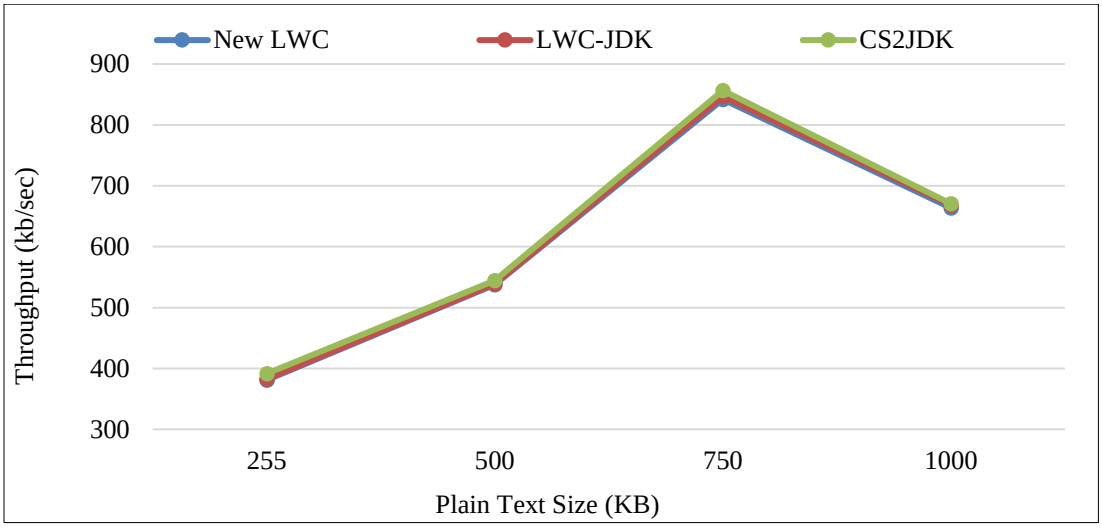


Fig. 9 Comparisons of throughput for various methods

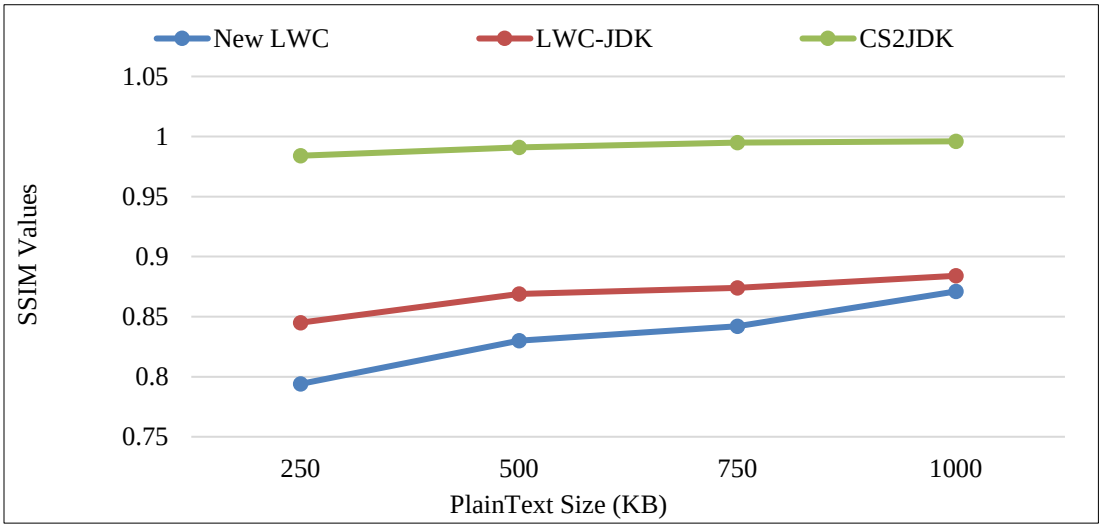


Fig. 10 Comparisons of SSIM values for various methods

Table 5. Comparisons of SSIM, PSNR & MSE

Plaintext Size (KB)	SSIM			PSNR (dB)			MSE		
	New LWC	LWC-JDK	CS2JDK	New LWC	LWC-JDK	CS2JDK	New LWC	LWC-JDK	CS2JDK
250	0.794	0.845	0.984	49.3	50.1	67.7	1.04	0.945	0.0074
500	0.83	0.869	0.991	54.2	54.83	68.3	1.03	0.69	0.0061
750	0.842	0.874	0.995	55.3	61.9	69.3	1.008	0.57	0.005
1000	0.871	0.884	0.996	57.2	62.64	70.8	1.007	0.46	0.0044

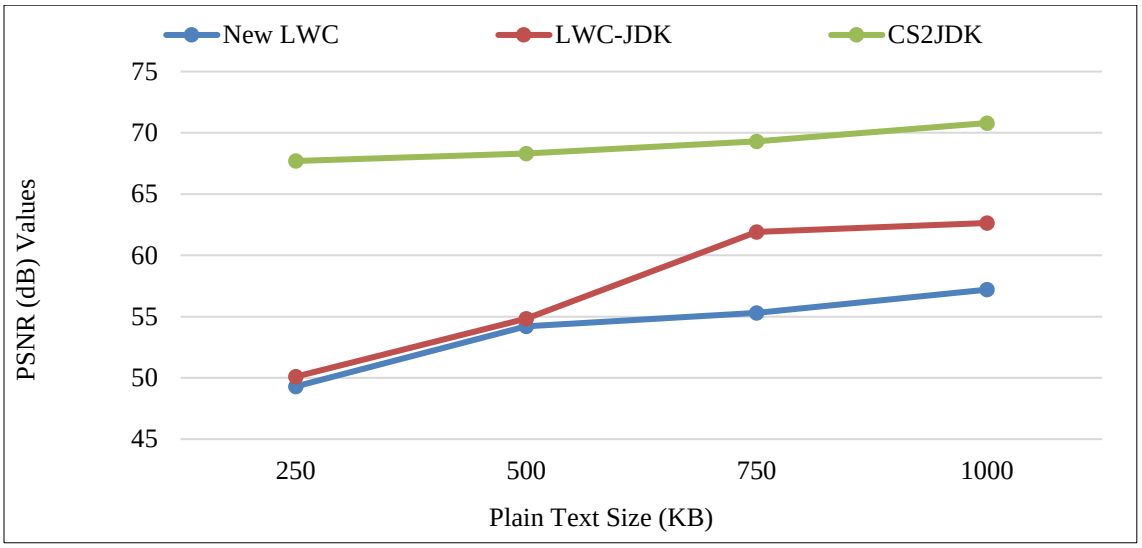


Fig. 11 Comparisons of PSNR values for various methods

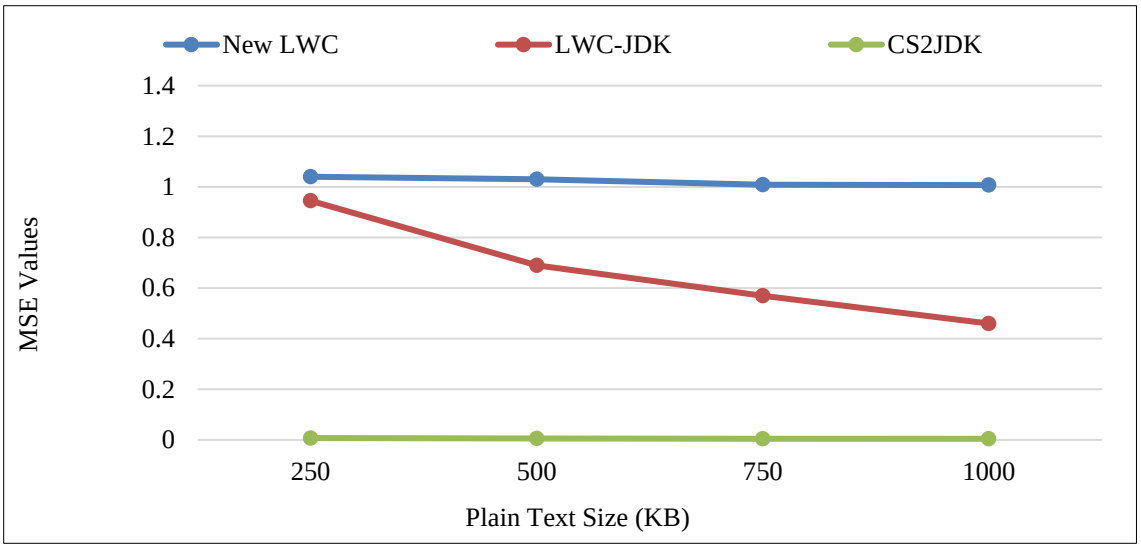


Fig. 12 Comparisons of MSE values for various methods

6. Conclusion

The appeal of cloud computing among businesses has increased because of its lower prices, fast re-provisioning, and distant connectivity. Users are excited about this novel computing paradigm but also concerned about the security risks that come with the cloud. One of the primary concerns surrounding CC these days is security. Numerous methods and approaches have been put forth, the most successful of which being cryptography. This paper proposes a new LWC procedure for improving data security in a CC environment called the constrained Spherical Scyphozoan jellyfish-based dynamic key (CS2JDK). It uses an enriched PCA-based compression scheme before transmitting the sensitive data. It made it possible to alter the cipher text in a particular pseudo-random hypothesis. This approach used modern Feistel architecture, which carries out encryption and decryption in a

manner akin to each other. Therefore, using the RWT module and optimizing key generation progress in integrated FB architecture makes a hybrid security system that leverages a 64-bit key with the most robust attack resiliency for multimedia data transfer over the cloud possible. The suggested security model's applicability for safe data transfer over CC environments while guaranteeing minimal computational overheads and complexity is confirmed by the qualitative and quantitative evaluation of the model. The suggested approach is strong enough to be utilized in an unpredictable cloud environment because it additionally guarantees various assault resiliencies. Emerging possibilities for safe data management in the industrial, military, medical, and other sectors may arise from the impact of cutting-edge technology like blockchain on improving data integrity and traceability in IoT.

References

[1] Fursan Thabit et al., "A New Lightweight Cryptographic Algorithm for Enhancing Data Security in Cloud," *Global Transitions Proceedings*, vol. 2, no. 1, pp. 91-99, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[2] Martin Koehler, and Siegfried Benkne, "VCE - A Versatile Cloud Environment for Scientific Applications," *The 7th International Conference on Autonomic and Autonomous Systems*, Venice/Mestre, Italy, pp. 81-87, 2011. [Google Scholar] [Publisher Link]

[3] Kiran et al., "Analysis and Computation of Encryption Technique to Enhance Security of Medical Images," *IOP Conference Series: Materials Science and Engineering: 1st International Conference on Computational Engineering and Material Science*, Karnataka, India, vol. 925, pp. 1-11, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[4] G.S. Pavithra, and N.V. Babu, "Energy Efficient Hierarchical Clustering Using HACOPSO in Wireless Sensor Networks," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 12, pp. 5219-5225, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[5] Shuang Liu, Li Liu, and Ming Pang, "Encryption Method and Security Analysis of Medical Images Based on Stream Cipher Enhanced Logical Mapping," *Technology and Health Care*, vol. 29, no. 1, pp. 185-193, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[6] B.D. Parameshachari et al., "Controlled Partial Image Encryption Based on LSIC and Chaotic Map," *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy*, Kuala Lumpur, Malaysia, pp. 60-63, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[7] Jaydip Sen, *Advances in Security in Computing and Communications*, IntechOpen, pp. 1-192, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[8] G. Viswanath, and P. Venkata Krishna, "Hybrid Encryption Framework for Securing Big Data Storage in Multi-Cloud Environment," *Evolutionary Intelligence*, vol. 14, no. 2, pp. 691-698, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[9] Saurabh Singh, Young-Sik Jeong, and Jong Hyuk Park, "A Survey on Cloud Computing Security: Issues, Threats, and Solutions," *Journal of Network and Computer Applications*, vol. 75, pp. 200-222, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[10] Aws Naser Jaber, and Mohamad Fadli Bin Zolkipli, "Use of Cryptography in Cloud Computing," *IEEE International Conference on Control System, Computing and Engineering*, Penang, Malaysia, pp. 179-184, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[11] Nelson Gonzalez et al., "A Quantitative Analysis of Current Security Concerns and Solutions for Cloud Computing," *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 1, no. 1, pp. 1-18, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[12] S. Gnana Sophia, K.K. Thanammal, and S.S. Sujatha, "Secure Cloud Medical Data Using Optimized Homomorphic Encryption," *Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 7, pp. 2702-2708, 2021. [Google Scholar] [Publisher Link]

[13] A. Mohamed Anwar, and S. Pavalarajan, "Spider Web-based Dynamic Key for Secured Transmission and Data-Aware Blockchain Encryption for the Internet of Things," *IETE Journal of Research*, vol. 70, no. 1, pp. 499-514, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] S. Balakrishnan, and K. Vinoth Kumar, "Hybrid Sine-Cosine Black Widow Spider Optimization Based Route Selection Protocol for Multihop Communication in IoT Assisted WSN," *Technical Bulletin*, vol. 30, no. 4, pp. 1159-1165, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] Khalil Hariss et al., "Design and Realization of a Fully Homomorphic Encryption Algorithm for Cloud Applications," *International Conference on Risks and Security of Internet and Systems*, Dinard, France, pp. 127-139, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[16] R. Udendhran, "A Hybrid Approach to Enhance Data Security in Cloud Storage," *2nd International Conference on Internet of things, Data and Cloud Computing*, Cambridge, United Kingdom, pp. 1-6, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[17] Eric Henziger, and Niklas Carlsson, "The Overhead of Confidentiality and Client-Side Encryption in Cloud Storage Systems," *Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing*, Auckland, New Zealand, pp. 209-217, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[18] Muhammad Wito Malik et al., "Development of Medical Image Encryption System Using Byte-Level Base-64 Encoding and AES Encryption Method," *Proceedings of the 6th International Conference on Communication and Information Processing*, Tokyo, Japan, pp. 153-158, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[19] Vishal Prakash, Ajay Vikram Singh, and Sunil Kumar Khatri, "A New Model of Light Weight Hybrid Cryptography for Internet of Things," *3rd International conference on Electronics, Communication and Aerospace Technology*, Coimbatore, India, pp. 282-285, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[20] Wang Ji Jun, and Tan Soo Fun, "A New Image Encryption Algorithm Based on Single S-Box and Dynamic Encryption Step," *IEEE Access*, vol. 9, pp. 120596-120612, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[21] Chengjian Liu et al., "Ensuring the Security and Performance of IoT Communication by Improving Encryption and Decryption with the Lightweight Cipher uBlock," *IEEE Systems Journal*, vol. 16, no. 4, pp. 5489-5500, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[22] Aruna Kumari Koppaka, and Vadlamani Naga Lakshmi, "An Efficient and Secured Big Data Storage in a Cloud-based Environment Using Hybrid Cryptography Algorithm and Rivest, Shamir, Adleman Algorithm," *International Journal of Intelligent Engineering & Systems*, vol. 17, no. 1, pp. 525-535, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[23] Martin Johns, and Alexandra Dirksen, "Towards Enabling Secure Web-Based Cloud Services Using Client-Side Encryption," *Proceedings of the ACM SIGSAC Conference on Cloud Computing Security Workshop*, USA, pp. 67-76, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[24] Eduardo B. Fernandez, "A Pattern for a Secure Cloud-Based IoT Architecture," *Proceedings of the 27th Conference on Pattern Languages of Programs*, pp. 1-9, 2022. [Google Scholar] [Publisher Link]

[25] P. Kanchanadevi et al., "An Attribute Based Encryption Scheme with Dynamic Attributes Supporting in the Hybrid Cloud," *Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, Palladam, India, pp. 271-273, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[26] Zeesha Mishra, and Bibhudendra Acharya, "High Throughput Novel Architectures of TEA Family for High Speed IoT and RFID Applications," *Journal of Information Security and Applications*, vol. 61, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[27] Samir Ifzarne, Imad Hafidi, and Nadia Idrissi, "Homomorphic Encryption for Compressed Sensing in Wireless Sensor Networks," *Proceedings of the 3rd International Conference on Smart City Applications*, Tetouan, Morocco, pp. 1-6, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[28] Fursan Thabit et al., "Exploration of Security Challenges in Cloud Computing: Issues, Threats, and Attacks with their Alleviating Techniques," *Journal of Information and Computational Science*, vol. 12, no. 10, pp. 35-57, 2020. [Google Scholar] [Publisher Link]

[29] Abdulrazzaq H.A. Al-Ahdal, Galal A. AL-Rummana, and Nilesh K. Deshmukh, "A Robust Lightweight Algorithm for Securing Data in Internet of Things Networks," *Sustainable Communication Networks and Application*, pp. 509-521, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[30] Fursan Thabit et al., "A Novel Effective Lightweight Homomorphic Cryptographic Algorithm for Data Security in Cloud Computing," *International Journal of Intelligent Networks*, vol. 3, pp. 16-30, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[31] Jui-Sheng Chou, and Dinh-Nhat Truong, "A Novel Metaheuristic Optimizer Inspired by Behavior of Jellyfish in Ocean," *Applied Mathematics and Computation*, vol. 389, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[32] Xiangyu Li, and Hua Wang, "Adaptive Principal Component Analysis," *Proceedings of the SIAM International Conference on Data Mining*, pp. 486-494, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[33] Esau Taiwo Oladipupo, Oluwakemi Christiana Abikoye, and Joseph Bamidele Awotunde, "A Lightweight Image Cryptosystem for Cloud-Assisted Internet of Things," *Applied Sciences*, vol. 14, no. 7, pp. 1-27, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[34] Pradeep Suthanthiramani et al., "Secured Data Storage and Retrieval Using Elliptic Curve Cryptography in Cloud," *The International Arab Journal of Information Technology*, vol. 18, no. 1, pp. 56-66, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[35] Mohamed Abdel-Basset et al., "An Improved Artificial Jellyfish Search Optimizer for Parameter Identification of Photovoltaic Models," *Energies*, vol. 14, no. 7, pp. 1-33, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[36] Baichi Chen et al., "A Lightweight Symmetric Image Encryption Cryptosystem in Wavelet Domain Based on an Improved Sine Map," *Chinese Physics B*, vol. 33, no. 3, pp. 1-11, 2024. [CrossRef] [Google Scholar] [Publisher Link]