RESEARCH ARTICLE

INTERNATIONAL RESEARCH JOURNAL OF MULTIDISCIPLINARY TECHNOVATION

Asian Research Association

DOI: 10.54392/irjmt2641

Artificial Rabbits Optimized Fuzzy Elliptic Curve Signcryption for Secured Data Transmission in WSN

J. Paruvathavardhini [a, *], B. Sargunam [a]

[a] Department of Electronics and Communication Engineering, School of Engineering, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore- 641043, India

* Corresponding Author Email: vardhini.jpvphd@gmail.com

DOI: https://doi.org/10.54392/irjmt2641

Received: 04-02-2026; Revised: 19-05-2026; Accepted: 26-05-2026; Published: 08-06-2026

Abstract: Secure and energy-efficient data transmission is a significant challenge in Wireless Sensor Networks (WSN). Many energy-saving and security schemes already address this issue. However, these schemes fail to strike a balance between energy efficiency, performance, and security, making them unsuitable for WSNs with limited resources. This research proposes a new system for energy-efficient and secure data transfer in WSNs to enhance network reliability and lifetime. The scheme has two phases: forwarder node selection and secure data transfer. In the selection phase, a Gaussian Likelihood Censored Regression (GLCR) method selects the most efficient adjacent node as the forwarder node, based on factors such as energy, packet loss, and others. This continues until the sink node is reached, creating a reliable path. Once a path is set, secure data transfer occurs using a proposed Optimized Fuzzy Elliptic Curve-based Signcryption (OFECS) scheme. Here, the Artificial Rabbits Optimization Algorithm (AROA) fine-tunes Fuzzy Elliptic Curve (FEC) parameters like elliptic curve coefficients, cofactor, group order, generator point, and prime number. This process enhances cryptographic security while reducing computational overhead. The scheme is simulated using the NS2 tool. Finally, results show that GLCR-OFECS achieves a 99.3% Packet Delivery Ratio (PDR), 0.8 mJ of Energy Consumption (EC), 1.3 ms of End-to-End Delay (EED), 245 kbps of throughput, 512 seconds of network lifetime outperforming existing schemes. For 2000 data units, it achieves 98.57% data confidentiality, 97.74% data integrity, 26.1 MB space complexity, and 32 ms execution time, surpassing existing WSN schemes for data transfer.

Keywords: WSN, Secure Data Transmission, Gaussian Likelihood Censored Regression, Fuzzy Elliptic Curve, Artificial Rabbits Optimization Algorithm, Signcryption.

1. Introduction

WSNs are becoming a key technology in many fields, such as smart cities, healthcare, ecological maintenance, and industrialized computerization. These systems are composed of many separate sensor nodes with data transfer, processing, and detecting capabilities [1]. The primary challenge in WSNs is power consumption, as distributed sensors in unreachable areas can become difficult to replace, leading to energy holes. Due to sensor nodes' lower energy capacity, efficient routing algorithms in WSNs are essential for extending the lifespan of networks and conserving energy during radio implementations [2]. To reduce energy usage in WSNs, various procedures like Low-Energy Adaptive Clustering Hierarchy (LEACH) have been developed that make use of cluster, schedule, contention, or low-duty cycle-based algorithms. Two kinds of clustering strategies (homogeneous and heterogeneous) are available for cluster algorithms, which are popular routing approaches in WSNs for improving scalability and prolonging network lifetime. Homogeneous mechanisms [3] are implemented in networks with identical initial hardware features and energy, while heterogeneous mechanisms are simulated in networks with varying levels of energy and hardware. As the nodes have constrained energy, the maximum of it must be utilized effectively despite the issues and challenges of WSN [4]. Hence, to impress both energy conservation and secure routing within WSN, a better method called Combinatorial Stochastic Sampling Bat Optimization via Quantified Indexive Energy-Aware Cluster optimization (QIEAC-CSSBO) [5] was presented. But neither of these techniques mentions enhancing the sensor's processing ability to reduce power usage while performing these tasks.

Securing data transmission in WSNs is vital because of the sensitivity of the data collected and sent

[6]. Unauthorized access, tampering, or eavesdropping can compromise integrity, confidentiality, and data accessibility, leading to serious consequences in critical applications. WSNs are vulnerable to assaults like eavesdropping, Denial of Service (DoS), and data interception. As technology advances, so do security threats. Cryptographic techniques [7], including algorithms and ciphers, are implemented to ensure confidentiality, authentication, integrity, non-repudiation, and service quality. Two fundamental synchronous and asymmetric keys are two forms of cryptography, with the originator linked to a public and private key.

Digital signatures in cryptography use a secret key and a calculated value to reassure receivers. They use three algorithms: key generation, signing, and signature verification. Digital certificates and signcryption are cryptographic methods for secure internet data transmission, reducing costs and ensuring privacy [8]. To overcome these challenges, this work is motivated. This paper addresses the twin purposes of enhancing energy efficiency and securing data transmission in WSNs: (1) energy-efficient forwarder node selection [9] and (2) secure data transmission [10]. The novelty of this study is that the AROA is successfully integrated to achieve an optimal balance between cryptographic security and computational efficiency of FEC parameters in resource-constrained WSN settings. The following is an overview of the work's primary contributions:

- In this article, GLCR-OFECS is proposed for energy-efficient and protected data transfers.
- First, GLCR is adopted to choose an efficient forwarder node for path formation between source and destination nodes. In this scheme, each node finds the most energy-efficient neighbor node as a forwarder based on different factors, such as data forwarding strategy, EC, residual energy, link quality, packet loss, distance to the sink, and data forwarding rate.
- Then, the OFECS scheme is suggested for protected data transfer. Here, the AROA is used to optimize the FEC parameters and enhance data confidentiality during communication with low computation overhead. The proposed scheme also helps achieve a tradeoff between efficiency and security in WSNs.
- Extensive simulations reveal that GLCR-OFECS outperforms conventional secure data transmission schemes in WSNs.

The structure of the research is designed as follows: Section 2 analyzes prior works on improving energy efficiency and protecting data transmission in WSNs; Section 3 presents a method; Section 4 examines the findings and debates; and Section 5 concludes the research with recommendations for future work.

2. Related Work

Recent years have perceived significant numerous research conducted in the energy-efficient node selection with secure transmission in WSN. This section presents a thorough literature assessment of earlier studies with the goal of identifying and addressing significant research obstacles.

Kumar *et al*. [11] introduced a novel approach for energy-efficient data transmission using an Improved Deep Convolutional Neural Network (IDCNN). The current phase of EE data transmission is when the Extended K-Means (EKM) technique is adopted. A t-Distribution-based Satin Bowerbird Optimization (t-DSBO) model will choose the cluster head automatically for everyone, depending on the nodes' leftover energy. However, wrong data will cause misidentification of such nodes in this process. Kalburgi *et al.* [12] presented a Taylor-Spotted Hyena Optimization Algorithm (TSHO) for electing head nodes in WSNs. TSHO is used to search for energy-efficient yet trustworthy nodes for data transmission. This can further lengthen the living time of the network as well as the overall efficiency. The proposed technique refers to trust-based data routing but does not go into more details of a specific protocol nor its potential impact in terms of the additional overhead introduced in the network. Vijayalakshmi *et al*. [13] suggested an energy-efficient adaptive cluster-head assortment system in WSN. This can greatly encompass the network lifespan of WSNs by reducing energy depletion of sensor nodes, the prime consideration in WSNs. Unlike the static approaches, most adaptive algorithms rely on extra computations. Depending on the complexity of the algorithm, such a cost may result in overhead-consuming energy on sensor nodes.

Qureshi *et al.* [14] presented an innovative Fuzzy Crow Search Optimization algorithm (F-CSO) to optimize various security parameters involved in data transmission. Such optimization may create a stronger defense mechanism. The introduction of fuzzy logic and optimization algorithms normally requires extra processing on nodes. This processing overhead may affect the battery life, especially in sensor nodes that are resource-constrained. Arya *et al.* [15] had presented a Deep Belief Network (DBN) based routing protocol in WSNs. Deep learning could possibly address complicated dynamics within networks and learn effective routing strategies, providing data transmission efficiency improvement. More requirement considerations are necessary for DBN security in WSNs.

Hu *et al*. [16] presented a Deep Reinforcement Learning (DRL) driven energy-efficient strategy in WSN. DRL utilizes deep learning models to familiarise its safety strategy in real-time based on shifting strategies for attack and network circumstances. This allows for a more dynamic and responsive security posture. The suggested method can be computationally expensive to train and run.

Table 1. Comparative Literature Review Analysis of Proposed Study

Ref. No.	Objective	Methodology	Key Findings
[Kumar *et al.*, 2021] [11]	Detecting malicious node for energy-efficient data transfer	1DCNN for malicious node detection, EKM for clustering, and t-DSBO for CH selection	It effectively detected the malicious node and rendered energy efficient data transmission.
[Kalburgi *et al.*, 2022] [12]	Mitigating link failures for improving network performance	Taylor-SHO for CH selection and mod-kVDPR for data routing	It efficiently reduced delay and energy with increased throughput.
[Vijayalakshmi *et al.*, 2023] [13]	Enhancing data transmission for secure routing	EEACHS for effective data transmission	It achieved high energy efficiency and network lifespan.
[Qureshi *et al.*, 2022] [14]	Providing secure data transmission efficiently	F-CSO and OLSR	Efficiently detected malicious nodes with higher network throughput and PDR.
[Arya *et al.*, 2022] [15]	Achieving efficient data transmission to increase node reachability.	Energy-efficient DBN-based routing protocol with RL and MRFO algorithm	Increased network longevity and PDR with lower energy usage.
[Hu *et al.*, 2024] [16]	Enhancing energy efficiency and security performance of WSN	DeepNR, including a defense mechanism	It ensured the secure data transmission with higher throughput and network lifetime.
[Sharma *et al.*, 2023] [17]	Developing energy-aware routing protocol without malicious/failed nodes.	LEACH with GOA and OANN-EATSRA	It increased packet receiving rate and reduced delay efficiently.
[Jalili *et al.*, 2024] [18]	Enhancing fault tolerance and routing efficiency.	Fault tolerance using Markov chain analysis	It achieved more reliable data transmission with low energy usage.
[Alrabea *et al.*, 2022] [19]	Designing energy efficient WSNs	Energy-efficient routing with weighted reward function	It improved energy and performance tradeoff by transferring data among adjacent nodes.
[Jalili *et al.*, 2024] [20]	Energy-efficient CH selection for reducing energy usage	Feed-forward multilayer neural network	It accurately identified suitable positions for CHs and reduced energy usage.
[Qiqieh *et al.*, 2024] [21]	Enhancing data security by incorporating robust key agreement protocol and encryption scheme	DNA-based cryptographic security framework	It ensured data security with lower encryption and decryption time.
[Alzubi *et al.*, 2020] [22]	Designing robust cryptosystem according to algebraic geometric curves	Hermitian-based cryptosystem	It increased data security compared to elliptic curves.

This could be a challenge for resource-constrained sensor nodes, potentially impacting battery life. Sharma *et al*. [17] introduced a protocol that utilizes Artificial Neural Networks (ANNs) to enhance routing choices for energy efficacy. It can potentially encompass the WSN's lifetime by lowering the amount of energy used on data transmission. Designing and training effective ANNs can be complex, requiring expertise and potentially large datasets for training.

Jalili *et al*. [18] introduced a new Markov chain-based model for fault-tolerant chain routing that saves energy in WSNs and by which perspectively the network remains highly reliable due to path changes with some knowledge of possible failures. This study mostly emphasizes fault-tolerant collaborative routing lacking secure data transmission. Alrabea *et al*. [19] introduced an energy concept-based model meant to optimize energy usage for their task in WSNs. Furthermore, this

model is involved to ensure that efficient energy-aware task allocation is given precedence to resolve the unnecessary computation.

Jalili *et al.* [20] developed a new model for head choice reduced in engaged energy of WSNs using a neural network-based model in a set of mobile networks with cluster formations carrying residues of node energy. Although the machine learning process might improve scalability, it would require additional time and computational effort, making it less practical for limited-resource WSNs. Qiqieh *et al.* [21] proposed a health-data-protective security framework for cloud on DNA encryption. It is this unique idea to merge bio-cryptography for physical key encryption for much more security, but with high computational costs. Alzubi *et al.* [22] proposed a Hermitian curve-based cryptosystem for IoT security, leveraging the various mathematical structures for facilitating good key management. This step will extend the strength of encryption; nonetheless, it has a pressing execution overhead regarding time and key management.

Table 1 summarizes the protocols studied above by organizing key goals, protocols, and results, enabling easy comparison of the new ideas and discoveries emerging from modern research.

2.1 Problem Statement

The existing energy-saving node selection approaches in WSNs face significant challenges [11-14]. Premature battery run-off resulting from poor energy management practices among nodes causes a reduced lifespan of the network. Similarly, existing secure data transmission algorithms [23-26] suffer from many drawbacks. Most algorithms require excessive execution time due to complex encryption schemes, making them unsuitable for real-time data transmission. These issues should be solved to design more efficient and scalable encryption schemes for improving network lifespan and reliability during secure data transfer.

3. Proposed Methodology

This section describes the suggested strategy of GLCR-OFECS for safe and energy-saving data transfer in WSN. Figure 1 displays a recap of GLCR-OFECS. First, the WSN is constructed, followed by two major processes: energy-efficient forwarder node selection using GLCR and secure data transmission using OFECS. This proposed scheme can save energy and lengthen the network life efficiently while securing communication.

3.1 Network Model

It includes a set of uniform sensor nodes denoted as $N = \{n_1, \ldots, n_n\}$. After being deployed, these sensor nodes remain stable and are dispersed randomly over an area with a size of $R \times R$. Every node in the network can detect activity, gather data, process it, and share it with others. The limited battery power of the nodes is largely used up when transmitting and receiving data through their radio transceiver.

3.2 Forwarder Node Selection Using GLCR Algorithm

For secure and efficient data transfer in WSN, the node parameters such as packet forwarding strategy, data forwarding rate, packet loss factor, residual energy, availability factor, link quality, distance to the sink, and coverage for every sensor node are initially calculated. The explanations of these parameters are given below:

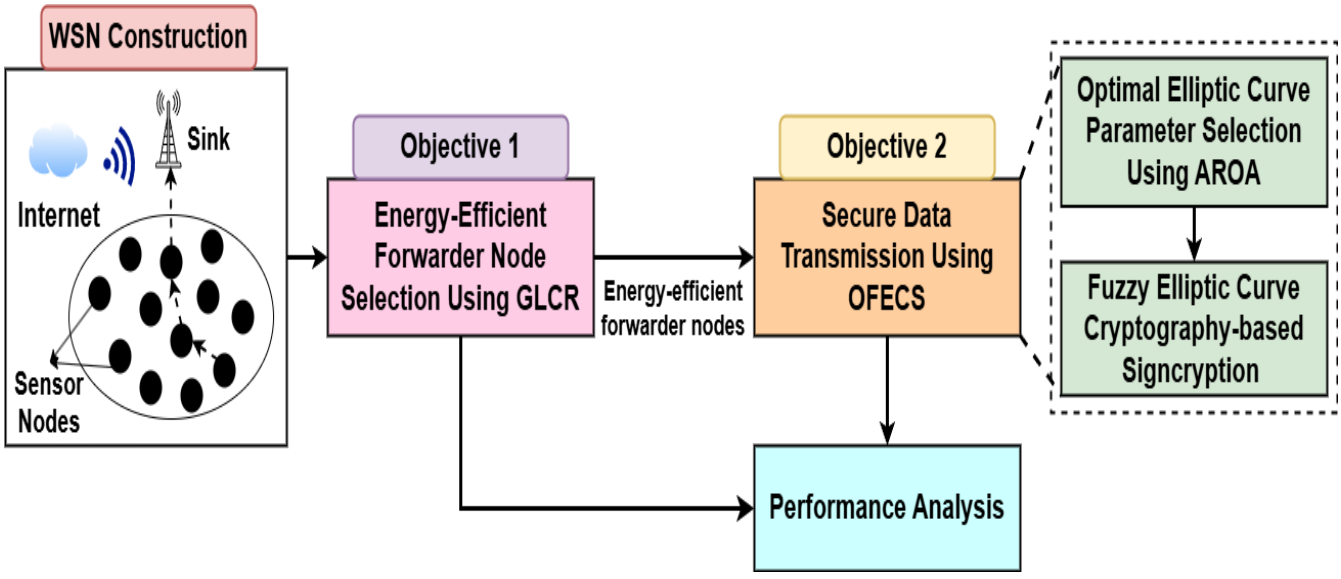

Figure 1. Workflow of Proposed GLCR- OFECS

3.2.1 Packet Forwarding Strategy

Packet forwarding strategy in WSNs involves determining the best route for transmitting data from the origin node to the sink. When data is dispatched through a higher energy node, the data may get lost and take more time (i.e., experience high delay). Therefore, it is essential to find the nodes with low EC in WSNs. To achieve this, a pair of WSN nodes is considered, namely, n_i and its neighbor noden_j. The data forwarding packet arrays at the initial stage are given in Eq. (1):

$$n_i \xrightarrow{[(D_{P1},\ldots,D_{PM})]} n_j \quad (1)$$

In Eq. (1), n_i is the i-th sensor node, n_j is the j-th adjacent node to n_i, and $(D_{P1},\ldots,D_{PM})$ denotes the list of packets that are communicated from n_i to n_j, where PM is the quantity of data packets.

3.2.2 Energy Consumption

The energy E of a WSN includes the sensor nodes' initial energy $E_{initial}$, energy used during transmissionE_T, energy used during data processingE_{DP}, energy consumed during receiving E_R, and energy consumed during the idle stateE_{idle}.

3.2.3 Link Quality

The link quality (LQ_{ij}) between n_i and n_j is computed as the strength of the expected packet, which is determined as:

$$LQ_{ij} \propto RSS \quad (2)$$

In Eq. (2), RSS is the received signal strength.

3.2.4 Data Forwarding Rate

The data forwarding rate in WSN is a crucial metric indicating the dependability and effectiveness of data transfer. In this scenario, each node avoids direct communication with the end node. If node n_i transmits data packets to the adjacent node n_j, then the data forwarding rate DFR_{ij} between nodes n_i and n_j is determined as:

$$DFR_{ij} = \begin{cases} \frac{S_T(n_i,n_j)-L_{lower}}{E_T(n_i,n_j)-L_{lower}}, & S_T(n_i,n_j) \le E_T(n_i,n_j) \\ \frac{L_{upper}-S_T(n_i,n_j)}{L_{upper}-E_T(n_i,n_j)}, & Or\ else \end{cases} \quad (3)$$

In Eq. (3), $S_T(n_i,n_j)$ is the data forwarding ratios at the initial period$T$, $E_T(n_i,n_j)$ denotes the anticipated number of packets forwarded at$T$, L_{lower} is the threshold lower limit, and L_{upper} is the threshold upper limit.

3.2.5 Residual Energy

Residual energy $(E_{residual})$ refers to the energy that remains after the transfer of data-forwarding packets which is:

$$E_{residual} = E_{initial} - E_{consumed} \quad (4)$$

In Eq. (4), $E_{consumed}$ denotes the overall energy expended by the node during transmission and reception.

3.2.6 Packet Loss Factor (PLF)

PLF is the proportion of missing data packets to the transmitted ones, and its equation is given in (5):

$$PLF = \frac{No.of\ packets\ lost\ at\ time\ t}{Total\ number\ of\ packets\ transmitted\ at\ time\ t} \quad (5)$$

3.2.7 Distance to the Sink

It is represented as the distance between any node $n_i, i = 1,\ldots,n$ and the sink. It is calculated as:

$$d_{i,sink} = \sqrt{(x_{sink}-x_i)^2 + (y_{sink}-y_i)^2} \quad (6)$$

In Eq. (6), (x_i, y_i) are the coordinates ofn_i, and (x_{sink}, y_{sink}) are the sink coordinates.

According to these factors, it is essential to find appropriate forwarding nodes for effective data transmission between source and sink nodes. For this purpose, this study introduces the GLCR algorithm, which helps in enhancing the network life by selecting nodes that consume energy efficiently, thus prolonging the overall lifespan of the network. GLCR was adopted for energy-saving node selection in WSNs because it can deal with problems of censoring in the substitute of analyzing sensor nodes with variable energy and transmission reliability. Unlike regression techniques, GLCR is effective for locating low-energy-consumption nodes through its modeling of partially observed data, focusing on maximizing the network life and minimizing premature discharge of the battery. Comparison studies with other machine learning techniques, such as DRL, ANNs, and F-CSO, show that although the above methods tend to enhance energy efficiency, they may incur unnecessary computational complexity and perhaps require extensive training data or take too long. GLCR provides computational efficiency without sacrificing predictive accuracy, making it very useful for WSN environments with scarce resources, where quick decision-making and energy conservation are vital concerns.

3.3 GLCR Algorithm

The GLCR is a statistical modeling technique used to analyze censored data in WSNs. It aids in efficient node selection by modeling and predicting network parameters. The framework collects data from

nodes, develops a model, handles censored data, estimates parameters, and chooses an appropriate forwarding node. For efficient node selection, GLCR collects and processes data from network nodes.

Consider a scenario where node n_i transmits a request packet to its adjacent node n_j and receives a reply packet. Node n_j then transmits a request packet to its adjacent node n_k and receives a reply packet, continuing this process until the sink node is reached. Each node collects information about other nodes, including data forwarding rate, remaining energy, packet loss factor, distance to the sink, link quality, and EC.

The observed data values for node n_i are denoted as x_i, resulting in $X = \{x_1, \dots, x_i\}, i \in n$. For GLCR, represent vectors x and matrices K with their elements in a steady form, i.e., x_i, K_{ij}. The transpose of x is denoted as $x^\top$. Assume $x \in \mathbb{R}^n$ is a vector of descriptive parameters, and $y^* \in \mathbb{R}$ is a response parameter. The problem involves learning a function $f: \mathbb{R}^n \to \mathbb{R}$ from a set $\mathcal{G} = \{x_1, \dots, x_n\}$ of n observations, where y is a changed version of y^*:

$$y = \begin{cases} l, & if\ y^* + \varepsilon \le l \\ y^* + \varepsilon, & if\ l < y^* + \varepsilon < u \\ u, & if\ y^* + \varepsilon \ge u \end{cases} \quad (7)$$

In Eq. (7), $l, u \in \mathbb{R}$ with $l < u$ are lower and upper threshold values, and $\varepsilon \sim \mathcal{N}(0, \sigma_{y_i}^2)$ is the Gaussian noise.

3.3.1 Prior Distribution

The latent values $y^* = f(x)$ can be described by the covariance matrix, which is the result of a zero-mean Gaussian process. Using the Gaussian kernel function $K_{ij} = k(x_i, x_j)$, the covariance matrix is calculated as:

$$k(x, x') = \sigma_f^2 e^{\left(-\frac{1}{2}\sum_{i=1}^{n}\frac{(x_i - x_i')^2}{l_i^2}\right)} \quad (8)$$

In Eq. (8), x_i is the i^{th} node parameter value x, and $\theta = \{\sigma_f^2, l_1, \dots, l_i\}$ is the set of hyperparameters, where σ_f^2 symbolizes the signal variance and l_i defines the variance of the function for the i^{th} node parameter of the explanatory variables.

For simplicity, the dependence on θ is neglected in the formulas. The multivariate Gaussian distribution before the function $\{f(x_i)\}$ is produced by the kernel function.

$$p(f) = \frac{1}{(2\pi)^{\frac{n}{2}}|K|^{\frac{n}{2}}} e^{\left(-\frac{1}{2}f^T K^{-1} f\right)} \quad (9)$$

In Eq. (9), K signifies the $n \times n$ covariance matrix with K_{ij}, which determines the similarity between node parameters.

3.3.2 Likelihood Distribution

To model correlations in the censored data from Eq. (10), an ideal likelihood for noise-free cases is defined as follows:

$$p(y \le \xi | f(x), l, u) = \begin{cases} 0, & if\ \xi < l \\ \Phi\left(\xi | f(x), \sigma_y^2\right), & if\ l \le \xi \le u \\ 1, & if\ \xi > u \end{cases} \quad (10)$$

In Eq. (10), $\Phi(x|\mu, \sigma^2)$ signifies the Cumulative Distribution Function (CDF) of the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$.

3.3.3 Posterior Distribution

The posterior distribution models the latent parameters based on the observed censored data. It is represented as:

$$p(f|X, y) = \frac{1}{Z} p(f|X) \prod_{i=1}^{n} p(y_i|f_i) \quad (11)$$

In Eq. (11), Z is the normalization term, which is determined by

$$Z = p(y|X) = \int p(f|X) \prod_{i=1}^{n} p(y_i|f_i)\, df \quad (12)$$

3.3.4 Expectation Propagation

If the posterior in Eq. (12) is not analytic, a deterministic approximate inference method called expectation propagation can help by estimating the probability using a local probability estimate using an unnormalized Gaussian function in the latent variable f_i, as:

$$p(y_i|f_i) \simeq t_i\left(f_i|\tilde{Z}_i, \tilde{\mu}_i, \tilde{\sigma_i}^2\right) = \tilde{Z}_i \mathcal{N}\left(\tilde{\mu}_i, \tilde{\sigma_i}^2\right) \quad (13)$$

In Eq. (13), t_i is the i^{th} point with parameters $\tilde{Z}_i$, $\tilde{\mu}_i$ and $\tilde{\sigma_i}^2$, which are simply abbreviated as $t_i(f_i)$. The product of the n likelihood approximations t_i is defined as:

$$\prod_{i=1}^{n} t_i\left(f_i|\tilde{Z}_i, \tilde{\mu}_i, \tilde{\sigma_i}^2\right) = \mathcal{N}\left(\tilde{\mu}, \tilde{\Sigma}\right) \prod_{i=1}^{n} \tilde{Z}_i \quad (14)$$

In Eq. (14), $\tilde{\mu}$ signifies the vector of $\tilde{\mu}_i$ and $\tilde{\Sigma}$ signifies a diagonal matrix with $\tilde{\Sigma}_{ii} = \tilde{\sigma_i}^2$. The posterior $p(f|\mathcal{G})$ is approximated by $q(f|\mathcal{G})$, which is

$$q(f|\mathcal{G}) = \frac{1}{Z} p(f) \prod_{i=1}^{n} t_i\left(f_i|\tilde{Z}_i, \tilde{\mu}_i, \tilde{\sigma_i}^2\right) = \mathcal{N}(\mu, \Sigma) \quad (15a)$$

$$\mu = \Sigma \tilde{\Sigma} \tilde{\mu} \quad (15b)$$

$$\Sigma = \left(K^{-1} + \tilde{\Sigma}^{-1}\right)^{-1} \quad (15c)$$

Consequently, the t_i approximations are updated consecutively by this approach. This is accomplished by merging the cavity distribution, which is the outcome of removing the i^{th} term from the approximate posterior, with the i^{th} exact likelihood term. In addition, t_i is updated using a Gaussian approximation to the non-Gaussian marginal. An example of a Gaussian distribution is the cavity distribution $q_{\setminus i}$.

$$q_{\setminus i} \propto \int p(f) \prod_{j \neq i} t_j(f_j) = \mathcal{N}(\mu_{\setminus i}, {\sigma_{\setminus i}}^2) \quad (16a)$$

Where

$$\mu_{\setminus i} = {\sigma_{\setminus i}}^2 ({\sigma_i}^{-2} \mu_i - {\tilde{\sigma}_i}^{-2} \tilde{\mu}_i) \quad (16b)$$

$${\sigma_{\setminus i}}^2 = ({\sigma_i}^{-2} - {\tilde{\sigma}_i}^{-2})^{-1} \quad (16c)$$

After that, the new unnormalized Gaussian marginal is obtained, which is the closest to the sum of the precise probability and the $q_{\setminus i}$ as:

$$\hat{q}(f_i) = \hat{Z}_i \mathcal{N}(\hat{\mu}_i, {\hat{\sigma}_i}^2) \simeq q_{\setminus i} p(y_i | f_i) \quad (17)$$

In Eq. (17), $\hat{Z}_i, \hat{\mu}_i$, and ${\hat{\sigma}_i}^2$ are the moments of node n_i , which are optimized by minimizing the Kullback-Leibler (KL) divergence such as $KL(p(x) \| q(x))$. These moments are determined as follows:

$$\tilde{\mu}_i = {\tilde{\sigma}_i}^2 ({\hat{\sigma}_i}^{-2} \hat{\mu}_i - {\sigma_{\setminus i}}^{-2} \mu_{\setminus i}) \quad (18a)$$

$${\tilde{\sigma}_i}^2 = ({\hat{\sigma}_i}^{-2} - {\sigma_{\setminus i}}^{-2})^{-1} \quad (18b$$

$$\tilde{Z}_i = \hat{Z}_i \sqrt{2\pi ({\sigma_{\setminus i}}^2 + {\tilde{\sigma}_i}^2)} exp\left(\frac{1}{2} \frac{(\mu_{\setminus i} - \tilde{\mu}_i)^2}{({\sigma_{\setminus i}}^2 + {\tilde{\sigma}_i}^2)}\right) \quad (18c)$$

3.3.5 Marginal Likelihood

The approximation of the expectation propagation system to the marginal likelihood in Eq. (16) is defined as follows to evaluate node parameters for energy-efficient node selection:

$$p(\mathcal{G}|\theta) = \int p(f|\theta_1) \prod_{i=1}^{n} p(y_i | f_i, \theta_2) df \approx \int p(f|\theta_1) \prod_{i=1}^{n} t_i(f_i) df \quad (19)$$

In Eq. (19), $\theta = \{\theta_1, \theta_2\}$ are the parameters with θ_1 representing covariance and θ_2 representing likelihood. These parameters are fine-tuned by minimizing the negative log marginal probability using gradient descent as:

$$\frac{\partial}{\partial \theta_1} \log p(\mathcal{G}|\theta) = -\frac{1}{2} \frac{\partial}{\partial \theta_1} (\log|\Sigma + K|) + \mu^T (\Sigma + K)^{-1} \mu \quad (20)$$

$$\frac{\partial}{\partial \theta_2} \log p(\mathcal{G}|\theta) = \frac{\partial}{\partial \theta_2} \sum_{i=1}^{n} \log \hat{Z}_i \quad (21)$$

According to this, the GLCR algorithm evaluates the marginal likelihood in Eq. (19) for each neighboring node n_j of node n_i using the observed parameter values. The node with the maximum marginal likelihood is referred to as the high energy-efficient node and chosen as the forwarding node. This procedure continues iteratively until the sink node is reached. Thus, the GLCR ensures the optimal node selection by maximizing energy efficiency and network reliability. After selecting all forwarder nodes between source and sink nodes, the reliable route is set up for data transfer. During data transfer, it is encrypted by the source and sent to sink, where data is decrypted to get the original data. This encryption and decryption for secure communication is performed using the OFECS scheme, which is explained in the section below.

3.4 Optimized Fuzzy Elliptic Curve-Based Signcryption for Secure Data Transmission

In this scheme, both public and private keys are generated for the source, forwarder, and destination nodes using OFECS. Its randomness is enhanced by using the FEC method. Fuzzy logic principles are applied during key generation to accommodate uncertainties and imprecision inherent in cryptographic operations. By leveraging fuzzy logic, the goal is to produce cryptographic keys with increased entropy and randomness, thereby augmenting the security and flexibility of the scheme against assaults. It intends to generate further robust and adaptive cryptographic systems capable of addressing complex and uncertain environments, ensuring heightened security in various applications, including secure communication networks and data encryption protocols. Figure 2 illustrates the process of secure data transmission between source and destination nodes using OFECS.

Figure 2. Secure Data Transmission using OFECS

The cryptographic domain parameters and operations in this section are explicitly defined as follows to make them reproducible and mathematically clear:

- Finite field: $\mathbb{F}_p$, where p represents a large prime.
- Elliptic curve: $E: y^2 = x^3 + ax + b$, where $a, b \in \mathbb{F}_p$ and $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$.
- Base (generator) point: $G \in E(\mathbb{F}_p)$ of order n.
- Group: $\langle G \rangle \subseteq E(\mathbb{F}_p)$, a cyclic subgroup of order n.
- Cofactor: $h = \frac{|E(\mathbb{F}_p)|}{n}$.
- Scalar multiplication: $Q = kG$, where $k \in \mathbb{Z}_n^*$.
- Hash functions: $H_1: E(\mathbb{F}_p) \to \{0,1\}^l$ and $H_2: \{0,1\}^* \to \mathbb{Z}_n$.

3.4.1 Artificial Rabbits Optimization Algorithm for Fuzzy Elliptic Curve Parameters Optimization

The FEC's parameters include the following:

- Choice of Elliptic curve: The curve's formulation $E: y^2 = x^3 + ax + b$ and specific constants a and b are curve coefficients that determine the network's security and efficiency.
- Field size: It is denoted by a prime number $\mathcal{p}$. The field size influences security and computational load. Larger fields improve security however they require vigilant balancing with efficacy.
- Generator point G: It is employed to define points (x, y) on the curve, which impacts the computational swiftness.
- Scalar multiplication: The function $Q = kP$, in which P signifies a point on the curve and $k \in \mathbb{Z}_n^*$ is a scalar. This is optimized for efficiency.
- Group order $\mathcal{n}$: It defines the number of points on the elliptic curve and plays a crucial role in the FEC system's security.
- Cofactor h: It defines the fraction between the number of points on the curve and $\mathcal{n}$. It is vital to describe the subgroup, which is utilized for cryptographic purposes.

The FEC scheme offers enhanced security with smaller key lengths during data transmission in WSNs. However, efficient FEC operation depends on the careful selection of variables, such as $(a, b, \mathcal{p}, G, \mathcal{n}, h)$. Improper parameter initialization can result in weak security or increased computational overhead.

Hence, optimal tuning of these parameters is essential to strike a balance between security and efficiency. In this study, AROA is utilized to optimally adjust the FEC parameters $(a, b, \mathcal{p}, G, \mathcal{n}, h)$, thereby achieving a balance between cryptographic strength and computational efficiency. AROA is a novel optimization model motivated by the survival strategy of rabbits. It starts by initializing a population of candidate solutions, where each solution represents a potential set of elliptic curve parameters. The process of using AROA to optimize the FEC parameters is outlined below. The key terms in the AROA for FEC parameter optimization are defined in Table 2.

The AROA involves four distinct steps: initialization, detour foraging, random hiding, and switching.

Step 1: Initialization

In the initial step of AROA, a population of N_r rabbits are arbitrarily generated in the search space, each defining a potential set of FEC variables $(a, b, \mathcal{p}, G, \mathcal{n}, h)$, i.e., $X_i = (a_i, b_i, \mathcal{p}_i, G_i, \mathcal{n}_i, h_i)$.

The initial location (values of $a_i, b_i, \mathcal{p}_i, G_i, \mathcal{n}_i, h_i$) of rabbit X_i is created by:

$$L_i = l + rand(u - l), i = 1, \dots, N_r \quad (22)$$

In Eq. (22), L_i is the location of the i^{th} rabbit, l and u are the lower and upper limits of the search space, N_r is the number of populations, and $rand$ is an arbitrary value ranging from 0 to 1.

Table 2. Descriptions of Basic Terms in AROA for FEC Parameter Optimization

Terms	Descriptions
Population of rabbits	The set of candidate solutions (rabbits) initialized in the search space; each represents a possible set of parameter values for the FEC.
Search space	A multidimensional space defined by the range of possible parameter values.
Fitness function	The objective function that assesses the security strength of parameter values in FEC for key generation, i.e., curve validation.

Step 2: Exploration (Detour foraging strategy)

In this stage, rabbits arbitrarily select nutriment out of their region for food. It is computed by:

$$L_i^{t+1} = L_i^t + R(L_i^t - L_\mathfrak{r}^t) + round\left(0.5 \times (0.05 + rand_1)\right) \times randn \quad (23)$$

$$R = c \times \left(\mathfrak{e} - e^{\left(\frac{t-1}{t_{max}}\right)^2}\right) \sin(2\pi \times rand_2) \quad (24)$$

$$c(k) = \begin{cases} 1, & if\ k == randperm(D) \\ 0, & otherwise \end{cases}, k = 1, \dots, D; D = 1, \dots, \lceil rand_3, D \rceil \quad (25)$$

In Eqns. (23-25), L_i^t is the location of the i^{th} rabbit at iteration t, L_i^{t+1} denotes the location of the i^{th} rabbit at iteration $t+1$, $L_\mathfrak{r}^t$ denotes the location of an arbitrarily chosen rabbit $\mathfrak{r}$ at t, R is a step size, c is a selection vector for dimensions, $\mathfrak{e}$ is a decaying factor, $rand_1, rand_2$, and $rand_3$ are arbitrarily chosen between [0,1], $\lceil\cdot\rceil$ signifies the ceil function, $round$ denotes the rounding to the nearest integer, t_{max} denotes the maximum number of iterations, $randn \in \mathcal{N}(0,1)$ is an arbitrary integer following a normal distribution, D is the problem dimension, $randperm$ defines a random permutation of integers from 1 to D, and k is the dimension index.

Step 3: Exploitation (Random hiding)

In this step, rabbits hide in their burrows and dig holes around them to evade capture. It is computed by:

$$L_i^{t+1} = L_i^t + R \times (rand_4 \times b_\mathfrak{r} - L_i^t) \quad (26)$$

$$b_\mathfrak{r} = L_i^t + \left(\left(\frac{t_{max}-t+1}{t_{max}}\right) \times rand_4\right) \times g \times L_i^t \quad (27)$$

$$g(k) = \begin{cases} 1, & if\ k = \lceil rand_5, D \rceil \\ 0, & otherwise \end{cases}, k = 1, \dots, D; j = 1, \dots, D \quad (28)$$

In Eqns. (26-28), $rand_4, rand_5$ are arbitrary values from 0 to 1, and $b_\mathfrak{r}$ is an arbitrarily selected reference (burrow) location for hiding, and g is a selection vector for dimensions.

Here, $\left(\left(\frac{t_{max}-t+1}{t_{max}}\right) \times rand_4\right)$ defines the hiding factor. Once the detour foraging and random hiding strategies are performed, AROA can select either the parent agent or child agent as follows:

$$L_i^{t+1} = \begin{cases} L_i^t, & f(L_i^t) \le f(L_i^{t+1}) \\ L_i^{t+1}, & f(L_i^t) > f(L_i^{t+1}) \end{cases} \quad (29)$$

In Eq. (29), $f(\cdot)$ is the fitness function. The fitness function for getting optimal parameter is described in algorithm3.

Step 4: Switching Strategy

Rabbits in the AROA will first use a detour foraging approach before switching to a random hiding strategy later. To transition between the two tactics, AROA uses the energy factor A. The rabbit has more energy and opts for the meandering foraging approach when $A > 1$. In contrast, the rabbit uses the random hiding technique when it lacks the energy to travel far. This energy factor A is computed by

$$A = 4 \times \left(1 - \frac{t}{t_{max}}\right) \times \ln\left(\frac{1}{rand}\right) \quad (30)$$

The pseudocode for FEC parameter optimization using AROA is summarized in Algorithm 1.

Algorithm 1. AROA for FEC Parameter Optimization

Input: Population size ($N_\mathfrak{r}$) and maximum iteration t_{max}

Output: Optimal $X_{best}(a, b, \mathcal{p}, G, \mathcal{n}, h)$ values

1. Begin
2. Initialize the population of rabbits $X_i, i = 1\ to\ N_\mathfrak{r}$ and its location L_i using Eq. (22);
3. $for(each\ rabbit\ X_i)$
4. $X_i \leftarrow$ GENERATE_ELLIPTIC_CURVE( ) //Function in algorithm 2
5. $f \leftarrow$ EVALUATE (X_i); // Function in algorithm 3
6. $end\ for$
7. Find the best rabbit X_{best} so far, which has the maximum fitness value;
8. $while(t < t_{max})$
9. Determine energy factor A using Eq. (30).
10. $for(each\ rabbit\ X_i)$
11. $if(A > 1)$
12. Select a rabbit arbitrarily from other individuals;
13. Determine R using Eq. (24);
14. Perform detour foraging using Eq. (23);
15. $f \leftarrow$EVALUATE(L_i^{t+1});
16. Update the location of current individual using Eq. (29);
17. $else$
18. Generate D burrows and arbitrarily select one as hiding using Eq. (27);
19. Perform random hiding using Eq. (26);
20. $f \leftarrow$EVALUATE (L_i^{t+1}););
21. Update the location of X_i using Eq. (29);
22. $end\ if$
23. Find the best solution X_{best};
24. $end\ for$
25. $end\ while$
26. Return X_{best}, i.e., optimal set of FEC parameter values;
27. End

The elliptic curves and their related variables are generated using Algorithm 2. It includes functions to initialize the prime numbers p and to determine the generator point G on the elliptic curve using mathematical operations, such as the Legendre symbol and the Tonelli-Shanks scheme. The Legendre symbol is utilized to confirm whether a value is a quadratic residue modulo$p$, which is crucial for identifying valid points on the elliptic curve. The Tonelli-Shanks method is applied to calculate the modular square root, enabling the calculation of the corresponding y-coordinate. These steps ensure that the generated points are valid and lie on the elliptic curve.

Algorithm 2. Initialization of FEC Parameters

1. function GENERATE_ELLIPTIC_CURVE
2. $while(True)$
3. Initialize p randomly within $[2^{k-1}, 2^k - 1]$, where k is the desired bit-length; //Here k=256
4. //Ensure p is prime
5. $if(p\ is\ not\ prime)$
6. Adjust p to nearest prime;
7. $end\ if$
8. $while(True)$
9. Generate a and b randomly in $[0, p-1]$;
10. //Check non-singularity condition
11. $if\left((4 \times a^3 + 27 \times b^2) \bmod p \neq 0\right)$
12. Break;
13. $end\ if$
14. $end\ while$
15. //Find generator point G
16. $for(x = 0 \text{ to } p-1)$
17. $rhs \leftarrow (x^3 + a \times x + b) \bmod p$;
18. //Check quadratic residue
19. $if\left(rhs^{\frac{p-1}{2}} \bmod p == 1\right)$
20. //Calculate y using Tonelli-Shanks method
21. Find y such that $y^2 = rhs \pmod p$;
22. $G \leftarrow (x, y)$;
23. Break;
24. $end\ if$
25. $end\ for$
26. $if(G\ exists)$
27. Break;
28. $end\ if$
29. $end\ while$
30. $n \leftarrow p - 1$;
31. $h \leftarrow 1$;
32. return FEC parameters (a, b, p, G, n, h)
33. end function

Fitness Function for AROA: The elliptic curve parameters are validated by verifying the primality of p, non-singularity of the curve, validity of the generator point G, and consistency of parameters such as n and h. These validations guarantee that the generated curve is suitable for secure cryptographic operations, such as key generation process. The fitness function evaluates each rabbit candidate based on curve validity, faithfulness to Hasse's theorem, and computational efficiency. A higher fitness value indicates a more suitable elliptic curve for FEC-based key generation. These validations and fitness function evaluations are presented in Algorithm 3.

The fitness function retrieves elliptic curve parameters (a, b, p, G, n, h). Curve validation returns 0 if the arguments do not form a curve. Curve order is computed and checked. The Hasse theorem boundaries and the Hasse score measure how close the order is to being predicted. A longer Pollard's rho attack execution time indicates stronger resistance. Also, an attack resistance score is determined. The final fitness calculation assigns 40% weight to the natural logarithm of the curve's order, 20% to the Hasse score, 20% to the attack's execution time, and 20% to resistance. The cumulative fitness score indicates elliptic curve appropriateness.

Algorithm 3. Evaluating Fitness Function of in AROA

1. function EVALUATE(a, b, p, G, n, h) //Function to evaluate the fitness of a rabbit in AROA
2. $if(\text{VALIDATE\_CURVE}(a, b, p, G, n, h) == 0)$
3. return 0
4. $end\ if$
5. $expected\_order \leftarrow p + 1 - 2 \times \sqrt{p}$;
6. $upper\_bound \leftarrow expected\_order + 2 \times \sqrt{p}$;
7. $lower\_bound \leftarrow expected\_order - 2 \times \sqrt{p}$;
8. $hasse\_score \leftarrow \max\left(0, \frac{upper\_bound - |n - expected\_order|}{upper\_bound - lower\_bound}\right)$;
9. $start\_time \leftarrow current\ time$;
10. $execution\_time \leftarrow current\ time - start\_time$;
11. $max\_time \leftarrow 10.0$;
12. $min\_time \leftarrow 0.1$;
13. $execution\_score \leftarrow \max\left(0, \min\left(1, \frac{max\ \_time - execution\_time}{max\ \_time - min\ \_time}\right)\right)$;
14. $rho\_attack\_result \leftarrow P\_RHO\_ATTACK(a, b, p, G, expected\ order)$;
15. $attack\_resistance\_score \leftarrow 1$ if rho_attack_result is None; otherwise 0;
16. $fitness\ (f) \leftarrow (0.4 \times \log(n)) + (0.2 \times hasse\_score \times \log(n)) + (0.2 \times execution\_score) + (0.2 \times attack\_resistance\_score)$;
17. return fitness (f);
18. end function
19. function VALIDATE_CURVE(a, b, p, G, n, h)
20. $if(h < 1)$
21. return 0;
22. $end\ if$
23. $if(p == 0)$

24. return 0;
25. *end if*
26. *if*$(length(G) == 2)$
27. $x, y \leftarrow G$;
28. *if*$\left((y^2 - x^3 - ax - b) \bmod p \neq 0\right)$
29. return 0;
30. *end if*
31. $field \leftarrow$ SubGroup with (p, G, n, h);
32. *if*$(no\ generator\ point\ in\ field)$
33. return 0
34. *end if*
35. $curve \leftarrow$ Curve with $(a, b, field, "random\_curve")$;
36. *else*
37. return 0
38. *end if*
39. $order \leftarrow n$;
40. *if*$(h \neq field.h)$
41. return 0
42. *end if*
43. *if*$\left((4a^3 + 27b^2) \bmod p = 0\right)$
44. return 0;
45. *end if*
46. *if*$(p \in [2,3]\ or\ p\ is\ not\ prime)$
47. return 0;
48. *end if*
49. *if*$(p == n)$
50. return 0;
51. *end if*
52. return 1
53. end function
54. function P_RHO_ATTACK$(a, b, p, G, order, max\ _iter)$
55. Initialize $Q_a \leftarrow G$, $Q_b \leftarrow G$, $a \leftarrow 0$, $b \leftarrow 0$;
56. $power\ of\ two \leftarrow 1$;
57. $iter \leftarrow 0$;
58. *while*$(iter < max\ \_iter)$
59. *for*$\left(in\ range(power\ of\ two)\right)$
60. $i \leftarrow Q_a[0] \bmod 3$;
61. *if*$(i = 0)$
62. $Q_a \leftarrow add\_points(Q_a, G, a, p)$
63. $a \leftarrow (a + 1) \bmod order$;
64. *else if*$(i = 1)$
65. $Q_a \leftarrow double\_and\_add(2, Q_a, a, p)$
66. $a \leftarrow (2 \times a) \bmod order$;
67. *else*
68. $Q_a \leftarrow double\_and\_add(2, Q_a, a, p)$
69. $a \leftarrow (2 \times a) \bmod order$;
70. $Q_a \leftarrow add\_points(Q_a, G, a, p)$
71. $a \leftarrow (a + 1) \bmod order$;
72. *end if*
73. *if*$\left(is\ distinguished(Q_a, t)\right)$
74. return a and Q_a
75. *end if*
76. *end for*
77. *for*$\left(in\ range(2)\right)$
78. Repeat the same steps for Q_b, but twice per iteration;
79. *end for*
80. $iter \leftarrow iter + 1$;
81. *if*$(Q_a = Q_b)$
82. $power\ of\ two \leftarrow power\ of\ two \times 2$;
83. $Q_b \leftarrow Q_a$;
84. $b \leftarrow a$;
85. *end if*
86. *end while*
87. return None
88. end function

Thus, the AROA optimizes parameters of elliptic curves used in FEC key generation to achieve an effective trade-off between security strength and computational efficiency. It reduces the computational overhead associated with key generation, making FEC more suitable for resource-constrained WSN environments. The computational complexity of AROA for optimizing FEC parameters is $O(t_{max} \times D \times N_r)$.

3.4.2 Fuzzy Elliptic Curve-Based Signcryption

The OFECS scheme is performed based on different phases: (i) setup, (ii) key generation, (iii) signcryption (i.e., encryption), and (iv) unsigncryption (i.e., decryption). Consider a data transmission with a source node (SN) and a destination node (DN). M_i signifies plaintext data in the form of characters or integers. K_i is selected from P, which acts as a shared parameter for both encryption and decryption processes. The working of each phase is explained in the following sections.

3.4.2.1 Setup

This is the system initialization method that necessitates safeguarding information of the algorithm's input, and the common parameters' output, denoted as λ. Consider $\mathbb{F}_p$ is a finite field of prime order p, E is an elliptic curve with length l, G symbolizes an elliptic curve cyclic multiplication set of order n generated by the base point of the elliptic curve P. If the plaintext space equals $\{0,1\}^l$, then dual hash functions are chosen such as $H_1: G \rightarrow \{0,1\}^l$ and $H_2: \{0,1\}^* \rightarrow Z_n$. So, parameters $\gamma = \{a, b, n, G, p, h\}$ and H_1 and H_2 are obtained.

3.4.2.2 Keygen

It means an algorithm for key generation that takes optimal $\{a, b, n, G, p, h\}$ from AROA and random number r as input. This random number can be generated from the Linear Feedback Shift Register (LFSR) in conventional elliptic curve cryptography techniques. However, this approach can impact network security and efficiency due to its randomness. Therefore, this study utilizes machine learning techniques such as fuzzy logic to generate high-entropy random numbers instead of pseudo-random numbers from LFSR.

3.4.2.2.1 Fuzzy Pseudo-Random Number Generator

The creation of U unique integers used to generate keys is assumed. Each of the u bins has an

arbitrary number between U and u of the bandwidth. Here U represents the range of generated integers and u denotes the number of bins ($u = 5$). Because of this, the fuzzy system produces an integer sequence in the interval $\{0,1, \dots, U\}$. Since $u = 5$, U is chose to be a huge integer that is a multiple of u. To determine which of the prior outputs can be used to generate the next output, a predetermined sampling pattern is employed. For each element $x_k[j]$, in the vector of u elements (bins) that makes up x_k, the amount of samples in the associated bin is represented to feed into the fuzzy scheme.

3.4.2.2.2 Input of Fuzzy Scheme

Assume a collection of prior outcomes generated by the fuzzy integer generator. On period k, the input to the fuzzy scheme is dependent on the most recent L (i.e., window size) data models. For every interval $j \in \left[(j-i)\,{}^{U}/_{u}, \left({}^{jU}/_{u}\right)\right]$ with a length of ${}^{U}/_{u}$. The input of the fuzzy system is denoted as $x_k[j]$, which corresponds to the amount of samples within the period j. The objective is to reduce the autocorrelation of the output sequence and avoid concentration in extreme intervals.

For every defined period or bin, represented as $\left[(i-1)\,{}^{U}/_{u}, ({}^{iU}/_{u})\right]$, 3 fuzzy groups are established: small (FzS), medium (FzM), and large (FzL). These groups define the categorization of the number of samples selected during period *I* as small, medium, or large. Consequently, the fuzzification process organizes the samples in $x_k[j]$ for every period j into a fit vector represented as $\left(FzS(x_k[j]), FzM(x_k[j]), FzL(x_k[j])\right)$. On period k, the outcome of the fuzzification process is represented by $I_z(k)$, which is described as:

$$I_z(k) = \begin{pmatrix} FzS(x_k[1]) & FzM(x_k[1]) & FzL(x_k[1]) \\ FzS(x_k[2]) & FzM(x_k[2]) & FzL(x_k[2]) \\ \vdots & \vdots & \vdots \\ FzS(x_k[u]) & FzM(x_k[u]) & FzL(x_k[u]) \end{pmatrix} \quad (31)$$

In Eq. (31), $I_z(k)$ defines the fuzzified input matrix of size $u \times 3$ capturing membership values across bins.

3.4.2.2.3 Fuzzy Rules

In a fuzzy generator, a fuzzy rule is established by: "IF x_k is SN_i THEN the output is DN_i."

$$\delta_i = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (32)$$

Here, SN and DN represent fuzzy state labels (e.g., Small (S), Medium (M), Large (L) fuzzy sets). It corresponds to the fuzzy Cartesian product $SN_i \times DN_i$, with SN_i represented as a 5×3 matrix δ_i. For instance, $SN_i = (M, L, L, S, L)$ corresponds to.

If the number of samples in period j is denoted as $x_k[j] = 5$, then $x_k[j]$ is classified as small, with a corresponding small membership value of $FzS(5) = 1/3$. The system evaluates all conditions in the IF part for each element of the vector x_k using the AND operator, yielding a value that may correspond to the minimum as described below:

$$Fz\delta_i(x_k) = \min_j\left(Fz\delta_i[j](x_k[j])\right) = \min_j\left(I_k {\delta_i}^T[j,j]\right) \quad (33)$$

The i^{th} rule provides the degree ω_{ik} as:

$$\omega_{ik} = Fz\delta_i(x_k) \quad (34)$$

The defuzzification process combines the results of the inference task to obtain the final outcome, using the center of gravity defuzzification method:

$$y_{out} = \frac{\sum_{l=1}^{U} x^l \omega_l}{\sum_{l \in U} \omega_l} \quad (35)$$

This y_{out} can serve as the random number r in the OFECS scheme. Thus, this Keygen algorithm takes γ and r as the input. Also, it yields a key pair (PK, SK) as an outcome.

The source and destination arbitrarily choose SK_S and SK_D as their private keys and determine their public keys as $PK_S = SK_S \times P$ and $PK_D = SK_D \times P$. The private keys SK_S and SK_D are chosen from random values r from a fuzzy random pseudo-generator. After that, both the source and destination nodes maintain their private keys secret and expose their public keys.

3.4.2.3 Signcrypt

This is a signcryption algorithm, which merges the digital signature and encryption to minimize the computational cost and communication overhead. It requires γ, the private key of the source SK_S, the public key of the destination PK_D, and data $\mathtt{M}$ as the input. Also, it requires signcryption τ as the output. The source node utilizes PK_D and SK_S to signcrypt the data $\mathtt{M}$ as follows:

1. Choose r from the fuzzy pseudo-random generator.
2. Calculate $r \times PK_D = Q$;
3. Calculate $b = H_1(Q)$;
4. Calculate ciphertext $c = b \oplus \mathtt{M}$; // $\oplus$ is bitwise XOR
5. Calculate hash-based signature $e = H_2(\mathtt{M}, Q, PK_S, PK_D)$;
6. Calculate digital signature $s = r^{-1}(e + SK_S)$. If $s = 0$, repeat this process from Step 1;
7. Obtain the signcryption $\tau = (c, e, s)$ and transmit it to the destination.

3.4.2.4 Unsigncrypt

This is an unsigncryption algorithm, which requires γ, the private key of the destination SK_D, the public key of the source PK_S, and τ as the input. It outputs the data M accepted or discarded (i.e., unsigncryption successful or failed). The destination obtains the signcryption $\tau = (c, e, s)$ and utilizes PK_S and SK_D to unsigncrypt it as follows:

1. Calculate $w = s^{-1}$;
2. Calculate $\mathcal{X} = (e \times w \times PK_D) + (w \times PK_S \times SK_D)$;
3. Calculate $b' = H_1(\mathcal{X})$;
4. Calculate plaintext $\mathrm{M} = b' \oplus c$;
5. Calculate $e' = H_2(\mathrm{M}, \mathcal{X}, PK_S, PK_D)$;
6. If $e' = e$, the destination node accepts the data M; otherwise, discard the data.

The entire process in the OFECS for data transmission in WSN is summarized in Algorithm 4.

Algorithm 4. Optimized Fuzzy-Based Elliptic Curve Signcryption

Input: Optimal $\mathrm{a}, \mathrm{b}, \mathcal{p}, \mathrm{G}, \mathcal{n}, \mathrm{h}$ values from the AROA
Output: Signcrypted data at the source and unsigncrypted data at the destination

1. Initialize $\mathbb{F}_{\mathcal{p}}$ of order $\mathcal{n}$;
2. Generate elliptic curve E using optimal values of $\mathrm{a}, \mathrm{b}, \mathcal{p}, \mathrm{G}, \mathcal{n}, \mathrm{h}$;
3. Define H_1 and H_2;
4. Procedure KEY DISTRIBUTION
5. Initialize source node S and destination node D;
6. S generates its private key $\mathrm{SK_S}$ and public key $\mathrm{PK_S}$ using a fuzzy pseudo-generator;
7. D generates its private key $\mathrm{SK_D}$ and public key $\mathrm{PK_D}$ using a fuzzy pseudo-generator;
8. S transmits $\mathrm{PK_S}$ to D;
9. D transmits $\mathrm{PK_D}$ to S;
10. End Procedure
11. Procedure SIGNCRYPTION AT NODE S
12. Generate r using the fuzzy pseudo-random generator;
13. Calculate $\mathrm{r} \times \mathrm{PK_D} = \mathrm{Q}$;
14. Calculate $\mathrm{b} = \mathrm{H}_1(\mathrm{Q})$;
15. Calculate $\mathrm{c} = \mathrm{b} \oplus \mathrm{M}$;
16. Calculate $\mathrm{e} = \mathrm{H}_2(\mathrm{M}, \mathrm{Q}, \mathrm{PK_S}, \mathrm{PK_D})$;
17. Calculate $\mathrm{s} = \mathrm{r}^{-1}(\mathrm{e} + \mathrm{SK_S})$. If $\mathrm{s} = 0$, repeat this process from Step 14;
18. Obtain the signcryption $\tau = (\mathrm{c}, \mathrm{e}, \mathrm{s})$ and transmit it to the destination;
19. End Procedure
20. Procedure UNSIGNCRYPTION AT DESTINATION NODE D
21. Calculate $\mathrm{w} = \mathrm{s}^{-1}$;
22. Calculate $\mathcal{X} = \mathrm{e} \times \mathrm{w} \times \mathrm{PK_D} + \mathrm{w} \times \mathrm{PK_S} \times \mathrm{SK_D}$;
23. Calculate $\mathrm{b}' = \mathrm{H}_1(\mathcal{X})$;
24. Calculate $\mathrm{M} = \mathrm{b}' \oplus \mathrm{c}$;
25. Calculate $\mathrm{e}' = \mathrm{H}_2(\mathrm{M}, \mathcal{X}, \mathrm{PK_S}, \mathrm{PK_D})$;
26. If $\mathrm{e}' = \mathrm{e}$, the destination node accepts the data M; otherwise, discard the data.
27. End Procedure

Hence, this study combines energy-efficient forwarder node selection with secure data transmission using GLCR-OFECS. It can enhance network security and lifetime while minimizing the EC, computation overhead, and communication overhead.

4. Result and Discussion

This segment assesses the efficiency of the GLCR-OFECS against conventional models, such as IDCNN-DSBO [16], Taylor-SHO [17], EEACHS [18], F-CSO [19], DBN [20], DRL [21], ANN [22], Advanced confinement of sets Deep learning Rabin certificateless sign Encryption (ADRES) [23] technique, Genetic Algorithm-based Elliptic-Curve Cryptography (GA-ECC) [24], Secure Encryption Random Permutation Pseudo Algorithm (SERPPA) [25] and Proxy Re-Encryption (PRE) [26].

4.1 Simulation Configuration

Table 3. Simulation Parameters

Parameters	Values
Monitoring zone	100×100 m²
Simulation Duration	4600 seconds (s)
Node energy level	0.1 Joule (J)
Number of SN	200
Transmission range	30 m
Number of rounds	3000
Transmission power	10 dBm
Network Model	Heterogeneous
Communication type	Single-hop and multi-hop
Packet size	512 bytes
MAC protocol	IEEE 802.11

The GLCR-OFECS method is implemented in the NS2 simulator on a system setup including 2 GB of RAM, an Intel Core processor, and the Windows 10 OS. Table 3 lists the variables employed in the simulation procedure. The current research is based on the assumption of a stationary WSN scenario, in which all sensor nodes are stationary once deployed. The nodes are randomly scattered in a monitoring area of 100×100 m², which is a typical stationary sensing application like environmental monitoring. The nodes are set to have different levels of energy to create realistic deployment conditions. The communication model enables one-hop and multi-hop data communication, based on the distance between nodes and the sink. It is assumed that

the wireless channel is ideal, i.e. external interference, fading of the signal, and hardware failure are not taken into account, and the routing efficiency and security performance are evaluated specifically. To measure robustness, predefined attack scenarios such as Brute-Force Attacks (BFA), DoS, and Man-in-the-Middle (MitM) attacks are included in the simulation. These attacks are conducted under controlled conditions to assess the effectiveness of the proposed OFECS scheme in ensuring data confidentiality and integrity.

All packets are expected to be of similar size to ensure uniformity in performance measurement. Additionally, the base station (sink node) is assumed to be secure and immune to attacks, serving as a trusted entity within the network. This simulation environment will offer a dependable and reproducible analysis of the proposed GLCR-OFECS framework by utilizing a fixed network model with consistent assumptions regarding communication, channel conditions, and attack scenarios.

4.2 Performance Comparison of Various Approaches

The efficiency of the GLCR-OFECS method is assessed using several key measures in this section. The measures include throughput, EC, EED, NLT, and PDR, which are evaluated and compared with the conventional methods namely IDCNN-DSBO [16], Taylor-SHO [17], EEACHS [18], and F-CSO [19]. Figure 3 shows the PDR analysis. The PDR is the quantity of packets that a node can transfer in a given amount of time. In the proposed method, the PDR stands at 99.3% with 100 nodes in the network. This rate slightly decreases to 90% when the network expands to 120 nodes. Typically, smaller networks exhibit higher packet rates.

However, as the network grows further, the delivery ratio rebounds, reaching 68.1% with 200 nodes. Despite this reduction, the overall PDR remains higher than the existing IDCNN-DSBO, Taylor-SHO, EEACHS, and F-CSO approaches, respectively. PDR refers to the proportion of packets successfully reaching their destination, with a higher value indicating superior protocol performance.

The EC analysis in Figure 4 demonstrates that the suggested GLCR-OFECS's energy usage increases as the nodes develop. The amount of resources used for sending and receiving data is known as the EC. Excessive packet collisions result in increased packet drop rates, leading to the need for additional energy expenditure to ensure successful packet delivery to the BS. However, GLCR-OFECS surpasses these challenges by minimizing error correction and preventing unnecessary transmissions. Existing approaches typically use more energy due to packet loss or crashes. At node 120, the GLCR-OFECS attains 0.015mJ, while existing IDCNN-DSBO, Taylor-SHO, EEACHS, and F-CSO achieve 0.2mJ, 0.25mJ, 0.74mJ, and 0.68mJ respectively.

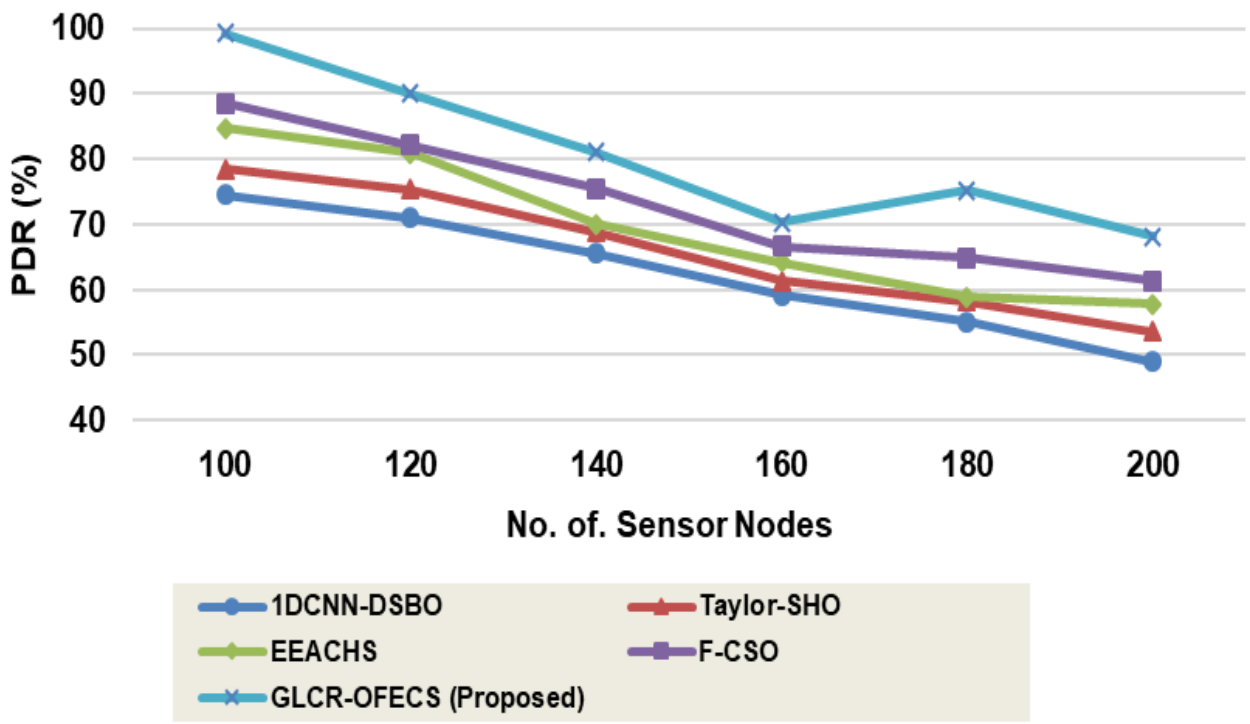

Figure 3. PDR analysis

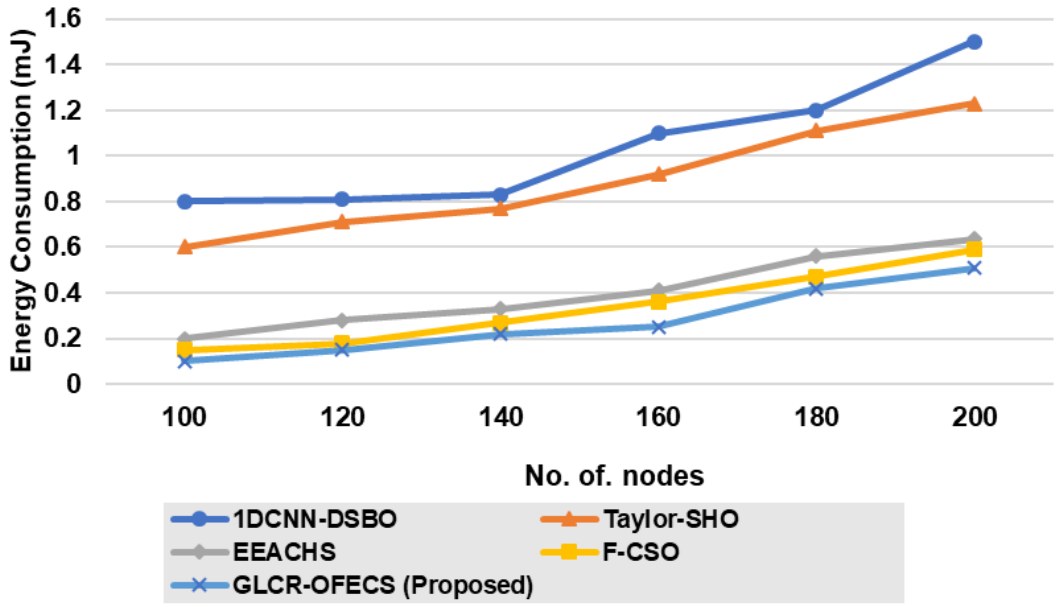

Figure 4. EC analysis

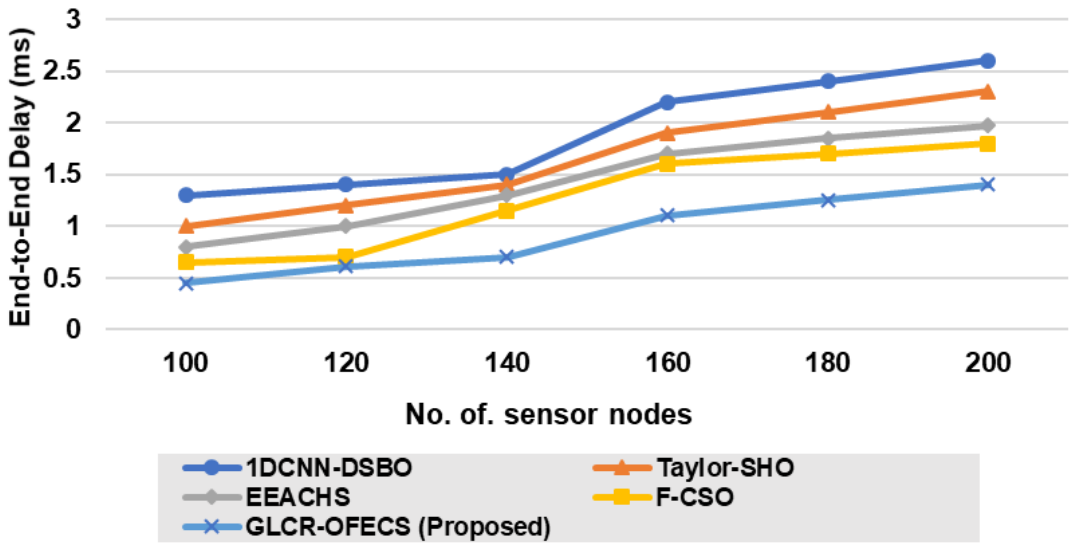

Figure 5. Delay Analysis

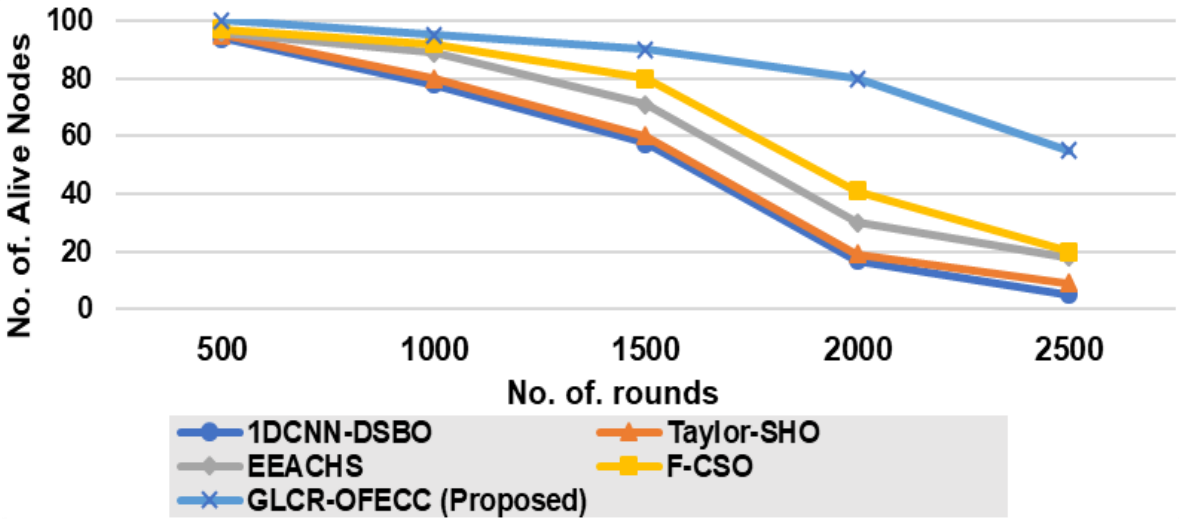

Figure 6. Alive Node Analysis

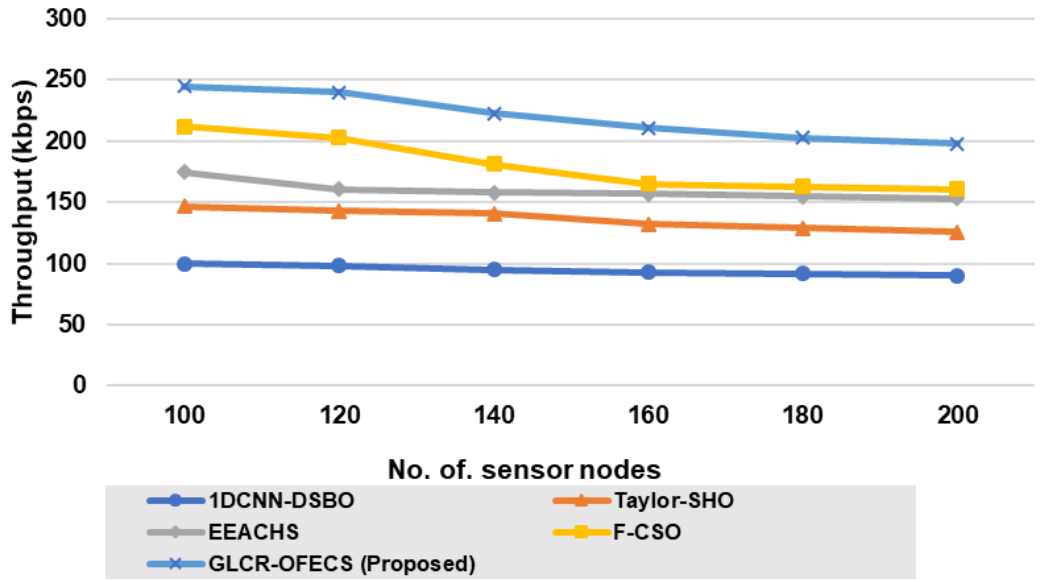

Figure 7. Throughput analysis

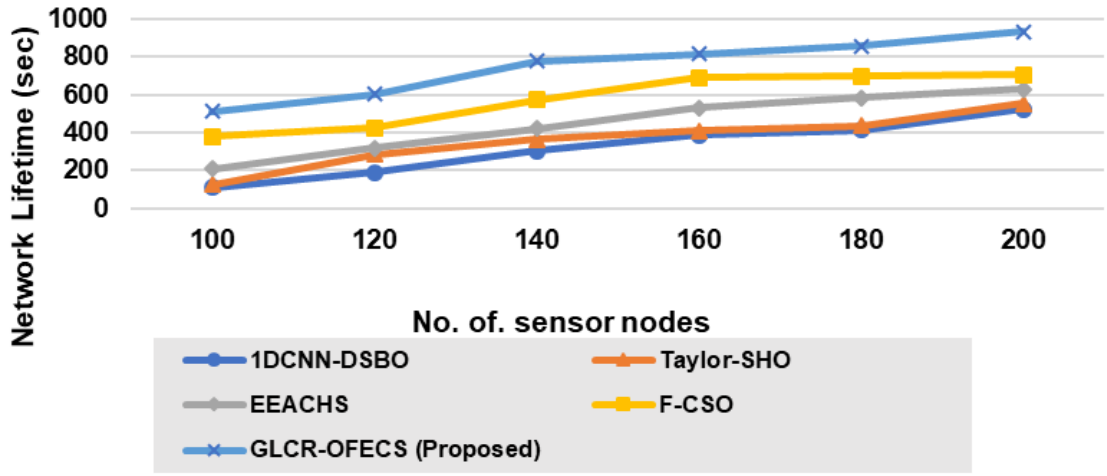

Figure 8. NLT analysis

Figure 5 depicts the delay analysis. The time it takes for every node to accept and retransfer a packet is denoted as EED. When the network has fewer nodes, the EED is minimal. But as the nodes increase, there is a slight growth in the EED. The average EED gives a typical duration spent by a packet to transport to its respective target. EED for the 100-node network is captured at a minimum of 0.5 ms, and the curve extends to 200 nodes. When the network is expanded to 200 nodes, the GLCR-OFECS EED is between 0.5ms-1ms. This means that with more nodes in the network, there is more EED.

Figure 6 illustrates the analysis of live nodes. Alive nodes are the active nodes in the network. From Figure 6, with a value of 48.01% at 3000 rounds, the GLCR-OFECS method achieves the highest number of alive nodes, thus a high NL. GLCR-OFECS uses the GLCR technique to move away from traditional paths in data communication to access the highest possible number of live nodes and avoid nodes getting stuck in local optima. Compared to 3000 rounds of existing IDCNN-DSBO, Taylor-SHO, EEACHS, and F-CSO algorithms, there were 0.210%, 0.278%, 0.345%, and 0.378% fewer live nodes.

The analysis of throughput is presented in Figure 7. Throughput refers to the average quantity of packet transmissions completed by a specific node. The number of packets available for throughput analysis is significantly influenced by the WSN's longevity and remaining power. Performance deteriorates as the number of nodes increases. Excessive network strain caused by increased traffic load reduces both network throughput and routing performance. At node 200, the proposed approach outperforms the current IDCNN-DSBO, Taylor-SHO, EEACHS, and F-CSO approaches in terms of throughput by 18.43%, 21.42%, 39.79%, and 23.46%, respectively.

The NLT analysis is shown in Figure 8. Network lifetime refers to the time taken for the network to

function efficiently before its energy reserves are depleted, affecting its ability to perform sensing, data processing, and communication tasks reliably. The NLT analysis in Figure 9 demonstrates that the recommended routing technique has a longer lifespan compared to other existing systems.

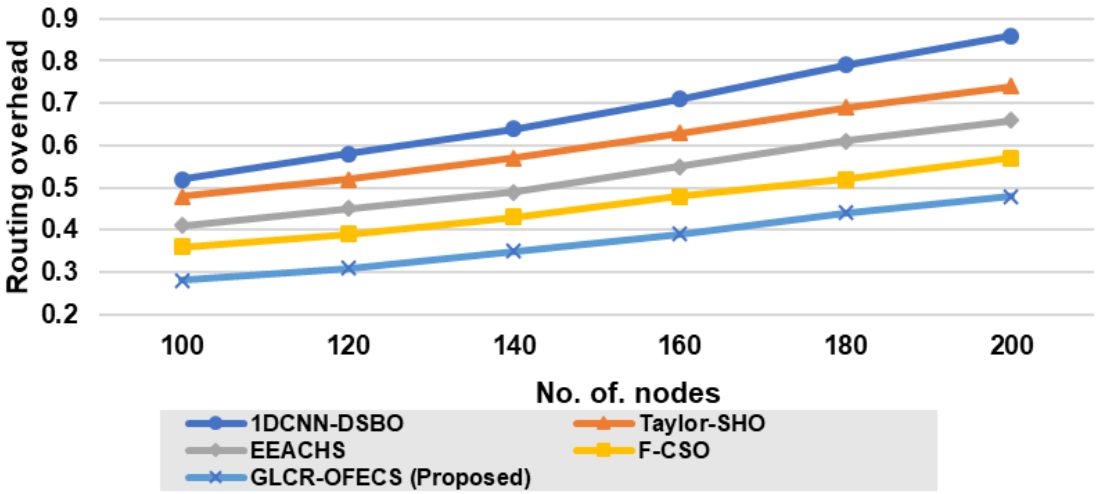

Figure 9. Routing overhead analysis

Table 4. Statistical Analysis for Network Performance

Metrics	Algorithms	Mean	Standard deviation	95% CI
PDR (%)	1DCNN-DSBO	62.42	9.35	[54.95-69.89]
	Taylor-SHO	65.97	9.20	[58.60-73.34]
	EEACHS	69.45	10.03	[61.43-77.47]
	F-CSO	73.17	10.21	[64.99-81.35]
	GLCR-OFECS	90.67	9.07	[71.83-99.51]
EC (mJ)	1DCNN-DSBO	1.04	0.26	[0.83-1.25]
	Taylor-SHO	0.89	0.22	[0.71-1.07]
	EEACHS	0.40	0.16	[0.27-0.53]
	F-CSO	0.34	0.17	[0.20-0.48]
	GLCR-OFECS	0.275	0.153	[0.15-0.40]
EED (ms)	1DCNN-DSBO	1.90	0.52	[1.48-2.32]
	Taylor-SHO	1.65	0.45	[1.29-2.01]
	EEACHS	1.44	0.41	[1.11-1.77]
	F-CSO	1.27	0.43	[0.93-1.61]
	GLCR-OFECS	0.918	0.386	[0.61-1.23]
Throughput (kbps)	1DCNN-DSBO	94.67	3.54	[91.84-97.50]
	Taylor-SHO	136.33	7.47	[130.36-142.30]
	EEACHS	159.83	7.76	[153.63-166.03]
	F-CSO	180.83	20.03	[164.82-196.84]
	GLCR-OFECS	220.00	18.62	[205.14-234.86]
Network lifetime (s)	1DCNN-DSBO	320.83	151.2	[199.6-442.0]
	Taylor-SHO	362.17	143.4	[247.5-476.8]
	EEACHS	449.17	158.6	[323.3-576.0]
	F-CSO	577.00	127.5	[475.0-679.0]
	GLCR-OFECS	748.67	160.32	[620.8-876.5]
Routing overhead	1DCNN-DSBO	0.68	0.12	[0.58-0.78]
	Taylor-SHO	0.61	0.10	[0.53-0.69]
	EEACHS	0.53	0.09	[0.46-0.60]
	F-CSO	0.46	0.08	[0.40-0.52]
	GLCR-OFECS	0.38	0.07	[0.32-0.44]

The proposed methodology enhances NLT due to its optimal route architecture and energy-efficient Main CH selection. Selecting the optimal EE node during data transmission helps prolong the NLT by conserving the energy of the SNs in the WSN. At 100 nodes, GLCR-OFECS has an NLT of approximately 190 seconds, while the other three techniques have network lifetimes of less than 160 seconds each, as each data aggregator node covers a limited area during EE node selection to avoid overlap.

Figure 9 shows the analysis of routing overhead. Routing overhead refers to the control packet cost incurred in the network to discover and maintain routes. The routing overhead is also increased by the increase in the number of nodes because of frequent route updates and increased network congestion. Such control traffic has a negative impact on the routing efficiency and performance. The proposed approach at node 200 decreases routing overhead by 44.19%, 35.14%, 27.27%, and 15.79% in comparison to IDCNN-DSBO, Taylor-SHO, EEACHS, and F-CSO, respectively.

The network performance assessment was repeated with different numbers of nodes (100 to 200) over 100 independent simulation runs to ensure statistical reliability. Stochastic elements like node distribution and communication patterns in each run were fixed for 10 random seeds. Table 4 reports the results such as the mean, standard deviation, and the 95% Confidence Intervals (CI). From these results, it is observed that the suggested GLCR-OFECS approach has a higher PDR and throughput, as well as reduced EC and EED, compared to current methods. The standard deviations are relatively small, and the confidence intervals are well-defined, which proves statistical stability and demonstrates that the improvements are not caused by random fluctuations under different network conditions.

4.3 Performance Comparison of Various Encryption Methods for Secure Transmission

For security analysis, the metrics Data Confidentiality Rate (DCR), Data Integrity Rate (DIR), Execution Time, and Space Complexity (SC) are assessed and contrasted with existing methods such as ADRES [23], GA-ECC [24], SERPPA [25] and PRE [26]. Figure 10 shows the DCR analysis. The DCR quantifies the proportion of data shielded from attacks (unauthorized entry) compared to the total data volume.

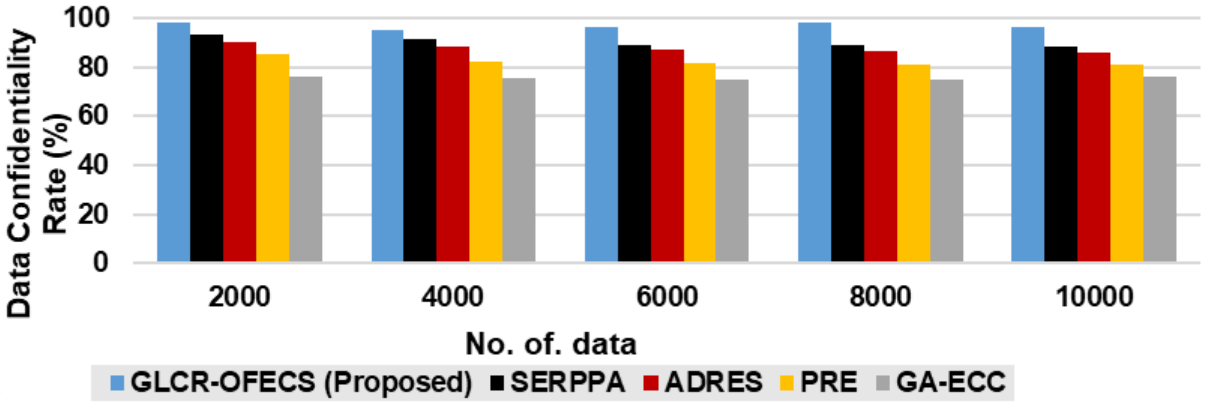

Figure 10. DCR analysis

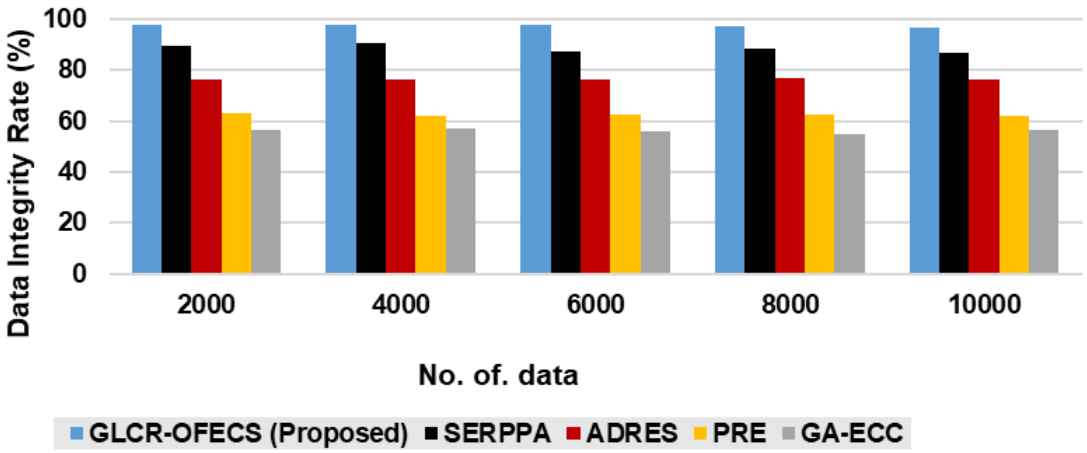

Figure 11. DIR analysis

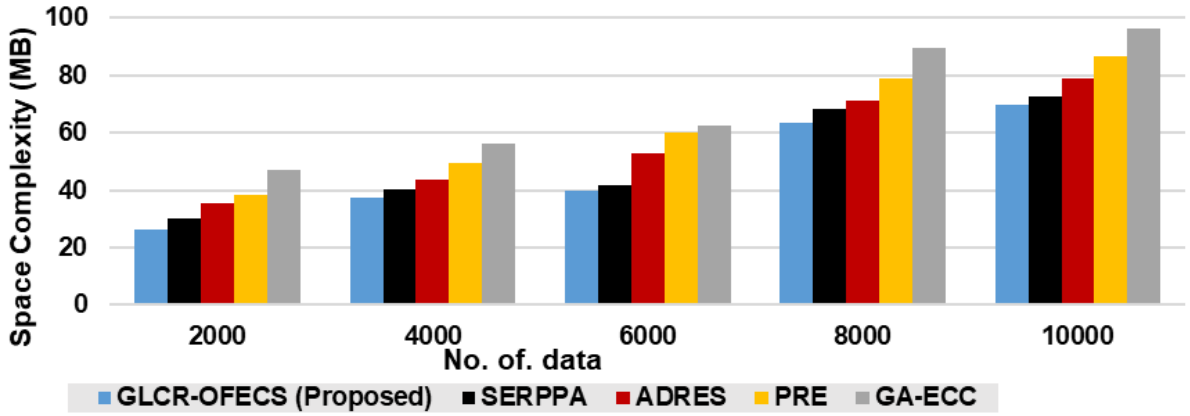

Figure 12. SC analysis

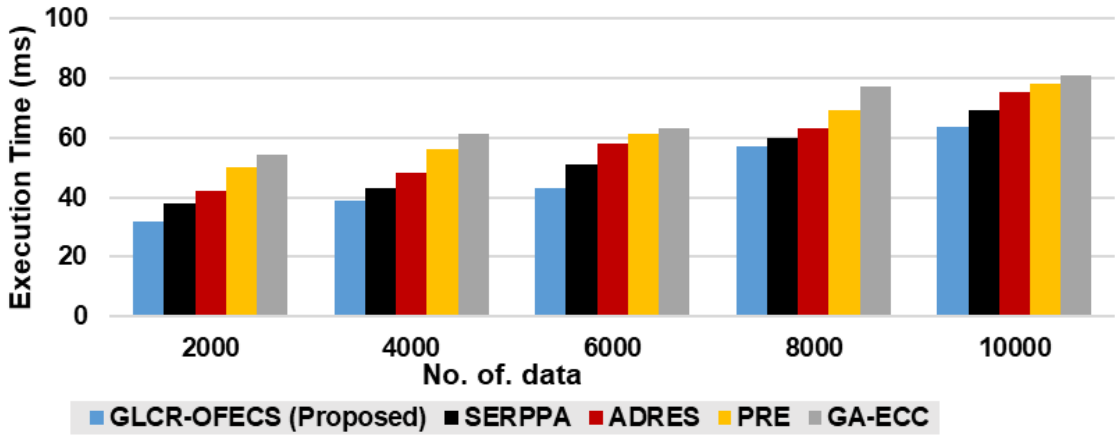

Figure 13. Execution Time analysis

$$DCR = \sum_{j-1}^{m} \frac{Number\ of\ data}{Number\ of\ protected\ data} \times 100 \quad (36)$$

These metrics gauge the effectiveness of data confidentiality measures. This can be formulated in equation (36), where j represents each individual data and m represents the total number of data.

In Figure 10, the GLCR-OFECS technique demonstrates enhanced DCR performance compared to existing methods. On average, it outperforms ADRES, GA-ECC, SERPPA, and PRE by 5%, 10%, 13%, and 12% respectively, as indicated by the four comparison results. Figure 11 shows the DIR analysis. DIR is defined as the proportion of data unaffected by unauthorized entry out of the total data volume. This can be formulated in equation (37):

$$DIR = \sum_{j-1}^{m} \frac{Number\ of\ data\ not\ altered}{Number\ of\ data} \times 100 \quad (37)$$

Based on the depicted Figure 11, the GLCR-OFECS technique demonstrates superior performance, yielding a high DIR in comparison to the existing approach. Specifically, it enhances the DIR by 5%, 7%, 18%, and 8% over ADRES, GA-ECC, SERPPA, and PRE respectively. This indicates that user data remained unaltered by attackers, thereby elevating the data integrity rate.

Figure 12 analyzes the SC. SC refers to the amount of memory required for storing data within sensor nodes. This can be formulated in Equation (38):

$$SC = \sum_{j-1}^{m} Number\ of\ data \times Space(data\ storing) \quad (38)$$

In the depicted Figure 12, the GLCR-OFECS method showcases lower SC compared to conventional approaches. On average, it reduces space complexity by 12%, 20%, 32%, and 24% compared to existing methods ADRES, GA-ECC, SERPPA, and PRE respectively. This suggests that GLCR-OFECS is more efficient in utilizing memory resources within the WSN context, thereby potentially offering benefits such as reduced storage overhead and improved scalability for data transmission and processing tasks in WSN environments. Figure 13 shows. Duration required by an OFECS algorithm to carry out secure data communication. This can be formulated in Equation (39):

$$ET = \sum_{j-1}^{m} Number\ of\ data \times Time(Secure\ data\ transmission) \quad (39)$$

The findings suggest that GLCR-OFECS significantly reduces execution time. On average, GLCR-OFECS decreases time usage by 10%, 20%, 32%, and 21% compared to ADRES, GA-ECC, SERPPA, and PRE, respectively.

The statistical analysis for network security performance was performed across varying data sizes (2000 to 10000) over 100 independent simulation runs and 10 random seeds. Mean, standard deviation, and 95% CI were used to summarize the results. The proposed GLCR-OFECS method demonstrated higher data confidentiality and integrity rates, as well as low space complexity and execution time, as shown in Table 5, compared to current methods. The standard deviation values of security metrics are relatively low, indicating a high level of consistency, and the confidence intervals are also small, further confirming the reliability of the results.

Table 5. Statistical Analysis for Network Security Performance

Metrics	Algorithms	Mean	Standard deviation	95% CI
Data confidentiality (%)	1DCNN-DSBO	62.42	9.35	[54.95-69.89]
	Taylor-SHO	65.97	9.20	[58.60-73.34]
	EEACHS	69.45	10.03	[61.43-77.47]
	F-CSO	73.17	10.21	[64.99-81.35]
	GLCR-OFECS	80.67	11.07	[71.83-89.51]
Data integrity (%)	SERPPA	88.61	1.62	[86.60-90.62]
	ADRES	76.47	0.33	[76.06-76.88]
	PRE	62.05	0.98	[60.83-63.27]
	GA-ECC	56.13	0.63	[55.35-56.91]
	GLCR-OFECS	97.29	0.50	[96.67-97.91]
Space complexity (MB)	SERPPA	50.64	18.32	[27.90-73.38]
	ADRES	56.36	17.54	[34.60-78.12]
	PRE	62.70	19.23	[38.80-86.60]
	GA-ECC	70.32	18.70	[47.05-93.59]
	GLCR-OFECS	47.34	18.28	[24.64-70.04]
Execution time (ms)	SERPPA	52.2	11.67	[37.72-66.88]
	ADRES	57.2	12.60	[41.55-72.85]
	PRE	62.8	11.13	[48.98-76.62]
	GA-ECC	67.2	10.78	[53.82-80.57]
	GLCR-OFECS	46.9	12.69	[31.20-62.60]

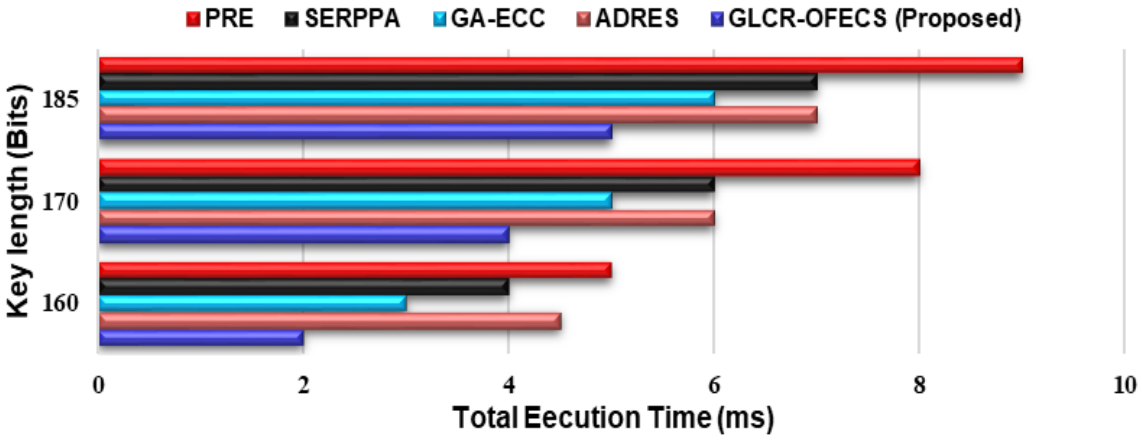

Figure 14. Total Execution Time analysis

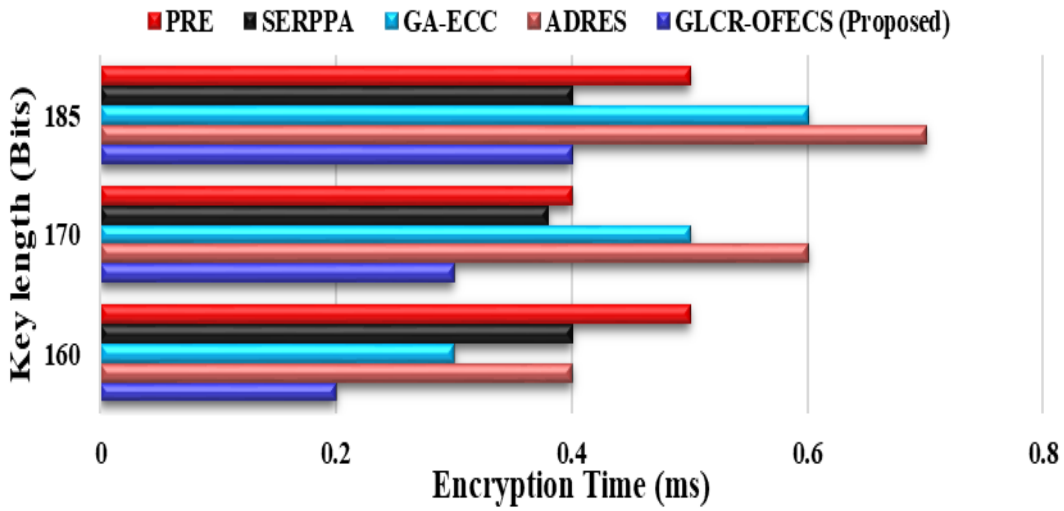

Figure 15. Encryption Time Analysis

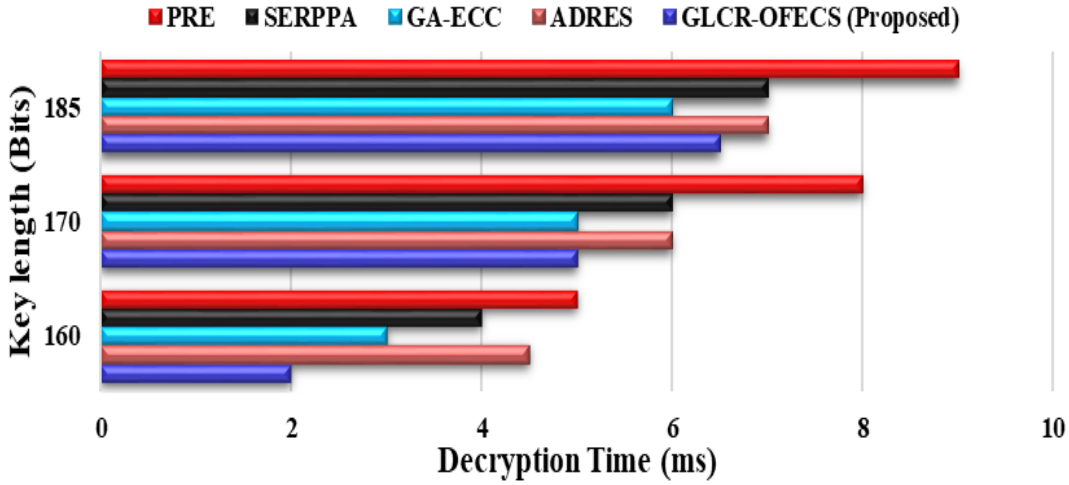

Figure 16. Decryption Time analysis

Additionally, the significant difference between the confidence limits of the proposed and baseline methods suggests that the improvements are statistically significant, supporting the effectiveness of the proposed method in ensuring security and computational efficiency.

In Figure 14, the total ET fluctuates based on the sizes of keys and text. Notably, ADRES, GA-ECC, SERPPA, and PRE exhibit longer runtime compared to the proposed system. As illustrated in Figure 15, encryption time escalates with key size. The proposed GLCR-OFECS method divides the plaintext into partitions, accelerating the encryption process compared to ADRES, GA-ECC, SERPPA, and PRE, respectively. The decryption time is observed to rise with the key size, as depicted in Figure 16. The proposed GLCR-OFECS approach partitions the cipher text, enhancing decryption speed compared to ADRES, GA-ECC, SERPPA, and PRE, respectively.

4.4 Security Analysis for Various Attacks

Formal threat model & adversary capabilities: Consider a set of honest nodes $U = \{U_1, U_2, \ldots\}$ and an adversary A. This adversary is assumed to capable of performing chosen-ciphertext attacks, replay attacks, and eavesdropping. The security properties are defined as follows:

Confidentiality: The adversary A selects two messages m_0 and m_1. A challenger encrypts one of them m_b, where $b \in \{0,1\}$, and A attempts to guess b. The confidentiality advantage is defined as follows:

$$Adv_A^{Conf} = \left| Pr[b' = b] - \frac{1}{2} \right| \quad (40)$$

Authenticity: It measures the probability that only valid signatures are accepted during verification. It is determined as:

$$Auth = \frac{Number\ of\ valid\ signatures\ accepted}{Total\ verification\ attempts} \quad (41)$$

Unforgeability: It defines the probability that an adversary successfully generates a valid signature without authorization as follows:

$$Adv_A^{forge} = Pr[successful\ forgery] \quad (42)$$

Replay resistance: It evaluates the network's ability to identify and discard replayed messages. It is computed as:

$$Replay_{rate} = \frac{Accepted\ replay\ messages}{Total\ replay\ attempts} \quad (43)$$

Chosen-ciphertext resistance: It is determined as:

$$Adv_A^{cca} = \left|Pr[b' = b] - \frac{1}{2}\right| \quad (44)$$

Table 6 presents a comparison of the security strength of the standard FECS and the proposed OFECS schemes in terms of various security properties. The findings show that OFECS is much more effective in enhancing security by decreasing the adversarial advantage from 2^{-128} to 2^{-160}, which translates to higher protection against brute-force and cryptanalytic attacks. Moreover, OFECS improves the detection of replay attacks (with more than a 99% detection rate) with insignificant replay acceptance rates. These advances can be largely credited to the optimization of elliptic curve parameter choice and better validation.

The proposed security system undergoes analysis against various attacks, such as BFA, DoS, and MitM. This assessment evaluates the system's security by considering Key Breaking Time (KBT) and the key similarity of compromised text. The analysis of the similarity of hacked text is outlined in Table 7.

Table 6. Security comparison between Standard FECS and Proposed OFECS

Security Properties	Standard FECS	Proposed OFECS
Confidentiality	$\leq 2^{-128}$	$\leq 2^{-160}$
Authenticity	$\approx 2^{-128}$	$\leq 2^{-160}$
Unforgeability	$\leq 2^{-128}$	$\leq 2^{-160}$
Replay resistance	≈ 0	≈ 0 (Detection > 99%)
Chosen-ciphertext attack resistance	$\leq 2^{-128}$	$\leq 2^{-160}$

Table 7. Key Similarity

Methods	Key Similarity (%)		
	DoSA	MitMA	BFA
GLCR-OFECS (Proposed)	13.5802	21.1111	13.7037
ADRES [23]	34.2222	30.5669	21.8544
GA-ECC [24]	38.4455	26.5622	18.1236
SERPPA [25]	18.7296	34.8718	23.5278
PRE [26]	23.1072	33.6322	25.4238

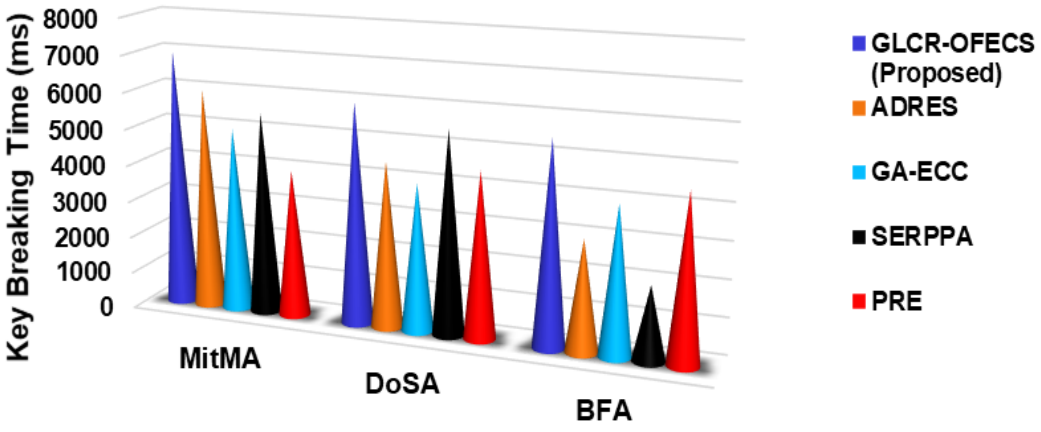

Figure 17. KBT analysis

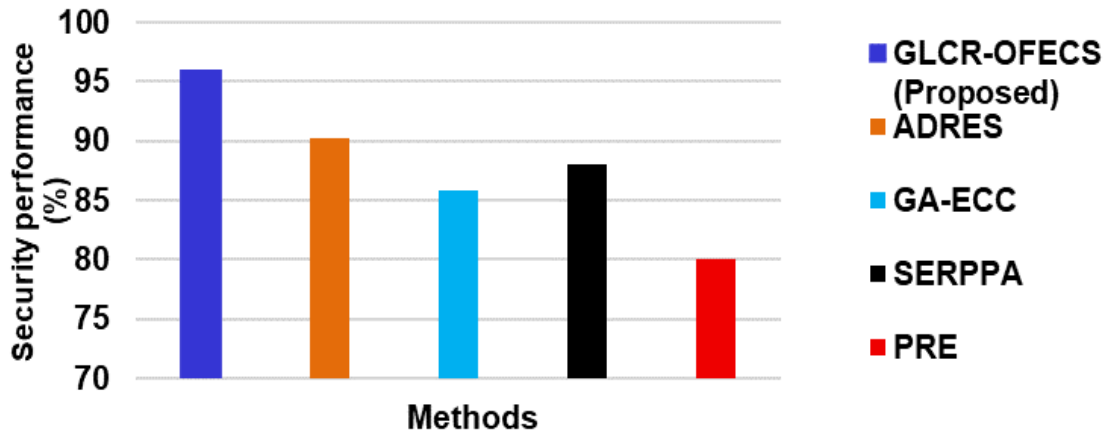

Figure 18. Security performance analysis

Table 8. Results for Ablation Study (200 nodes)

Algorithms	PDR (%)	EED (ms)	EC (mJ)	Throughput (kbps)	Network lifetime (s)
GLCR only	60.2	1.65	0.58	172	785
OFECS only	65.4	1.32	0.55	185	860
GLCR-OFECS	68.1	1.40	0.51	198	930

By utilizing a distance measure, the resemblance between the original and compromised text is evaluated. The table demonstrates that our proposed GLCR-OFECS achieves low similarity, indicating superior security compared to other encryption algorithms. Figure 17 illustrates the key breaking time for different encryption algorithms under various security attacks. The GLCR-OFECS algorithm demonstrates robust resilience against attacks, with the encryption key taking significantly longer to break. Notably, the key breaking time is notably superior for our proposed algorithm compared to others.

Figure 17 and Figure 18 shows the security performance. The presented approach indicates a security performance of 99% as per the security performance curve. Utilizing GLCR-OFECS methods, which incorporate OFECS, yields favorable results compared to existing techniques. The overall security performance demonstrates a notably high value of 99% when compared to ADRES, GA-ECC, SERPPA, and PRE, respectively. The 99% security improvement of the suggested GLCR-OFECS framework is measured with the help of key performance indicators like DCR, DIR, Key Similarity under Attacks, and Key Breaking Time (KBT). These KPIs assess the effectiveness of the system design against BFA, DoS, and others. Signature verification depending on the Jaccard Index strengthens authentication and makes unauthorized access risk lower, while AROA-optimized OFECS encryption enhances cryptographic strength against brute force decryption.

4.5 Ablation Study

This section introduces the ablation experiment that was carried out to compare the individual and combined roles of the GLCR routing mechanism and the OFECS security framework in transmitting data safely in WSNs. Table 8 provides the results of three configurations: GLCR-only, OFECS-only, and the combined model of GLCR-OFECS, under the same simulation conditions as outlined in Table 3. The comparison is conducted on a network size of 200 nodes based on key performance indicators, such as PDR, EED, EC, throughput, and network lifetime.

Based on Table 8, it can be seen that the performance of the OFECS-only setup is better than that of GLCR-only in both PDR (65.4%) and lower delay (1.32 ms) due to its effective and secure data transmission system. Conversely, the GLCR-only model has a greater delay (1.65 ms) and reduced PDR (60.2%), showing the constraints of routing without the integrated security optimization. GLCR, however, helps to stabilize routing and moderate EC. The proposed GLCR-OFECS model is better than the two configurations since it has the highest PDR (68.1%), throughput (198 kbps), and network lifetime (930 s), as well as the lowest EC (0.51 mJ). Despite the fact that there is a slight delay increase (1.40 ms) compared to the case of OFECS-only, the delay is still much lower than the case of GLCR-only, which indicates a balanced trade-off between efficiency and security.

Thus, the findings affirm that although OFECS increases the security and efficiency of transmission, and GLCR ensures the stability of routing, the

combination of both results in outstanding network performance, which proves the efficiency of the suggested solution.

4.6 Discussion

The proposed GLCR-OFECS method has been evaluated in the context of real-world attacks, namely BFA, DoS attacks, and MitM attacks. Simulation results suggest that OFECS greatly decreases key similarity in the compromised texts, thus making brute-force decryption exceedingly impractical. The Jaccard-based signature verification further augments this authentication process, countering spoofing and injection attacks. OFECS also attests to its resilience against DoS by improving encryption and decryption times to the minimum possible overhead to avert possible exhaustion on the part of the nodes. In terms of scalability, performance metrics including execution time, NLT, and PDR were evaluated concerning increasing network sizes. The results indicated that even with an increasing number of sensor nodes up to 200, OFECS still maintained high security, at 99%, with an equally shorter execution time. OFECS therefore supports WSN applications with robust security and energy efficiency, which is an improvement over other method options like GA-ECC, SERPPA, and PRE in terms of space complexity by up to 32% and throughput up to 39.79%.

The scalable and computationally efficient cryptographic framework specifically designed for WSNs is based on the combination of GLCR-OFECS. The proposed framework is more efficient than other traditional methods such as GA-ECC, SERPPA, and PRE. OFECS has optimized the elliptic curve parameters using AROA such that execution time becomes minimal, while strong encryption security is still upheld. The proposed OFECS increases the percentage of delivery in PDR by 99.3% and diminishes space complexity by 17%. Admittedly, the GLCR-OFECS makes huge improvements in PDR, energy usage, and delay; however, it still needs to be tested in more dynamic and large-scale WSN topologies (greater than 200 nodes). OFECS with AROA also assures cost-effectiveness in cryptography, but it is also concerned with real-time key management overhead in the motilities of WSNs. Some baseline results also reveal that there is an insignificant decline (from 99.3% when the node number is 100 to 68.1% at 200) in PDR as the nodes become more numerous, while EED becomes higher. For this, the fusing of hierarchical clustering, adaptive key management, and aware routing strategies have the potential to drive down the cost of operations and involve minimal latency but high security and energy efficiency beyond 200+ nodes or dynamic topologies. Future work will be directed to study a hybrid ML-driven routing and a blockchain-based authentication method for increased robustness in large-scale WSNs.

Using blockchain-based security frameworks within the proposed GLCR-OFECS system allows strengthening to a further extent security, trust, and efficacy in WSNs by providing decentralized authentication, tamper-proof data integrity, and secure key management. Sensitive data can be stored and accessed using a lightweight blockchain at the sink node or cluster heads, reduced to sensor node identities, encrypted transmissions, or authentication data, which eliminates the need for certificate authority. By creating smart contracts, secure node authentication and dynamic key distribution will be achieved by restricting it only to specified participants and blocking Sybil and replaying attacks. Finally, integrating Proof-of-Authority (PoA) or Delegated Proof-of-Stake (DPoS) consensus mechanisms will minimize computation cost overhead and thus make blockchain applicable to resource-constrained WSNs. Therefore, by combining blockchain with OFECS, the system can achieve trustless security, energy-efficient key management, and increased resilience against attacks while providing it with extensibility to scale up for large-scale IoT and industrial WSN. The specific context of the application, constraints on the network configuration, and exterior architecture of risk assumptions will ultimately determine energy security trade-offs in WSNs. For highly mission-critical applications where security breaches could endanger human life and breach sensitive data such as healthcare monitoring, military surveillance, and industrial automation security takes precedence at the expense of increased EC. Where security is less at risk and has stringent energy constraints such as environment monitoring or smart agriculture, energy efficiency becomes the focus to encompass the network life and hence long-term functioning. Ideally, new networks should be balanced: lightweight cryptographic protocols along with adaptive mechanisms and energy-aware routing could minimize power consumption while providing a significant security guarantee. Such a complement can help achieve the long life of the network while giving it strong security based on real-time operational needs "dynamic security adaptation, hardware-accelerated encryption, and machine learning-driven anomaly detection". The proposed GLCR-OFECS method can be adopted in a variety of applications involving WSN, such as healthcare monitoring and industrial IoT, fine-tuning node selection and security mechanisms to fit application-specific constraints when working on aspects of energy efficiency. In healthcare WSNs that mainly deal with patient data privacy while requiring real-time monitoring, low-latency encryption, and certificate authentication can support the secure transmission of electrocardiogram (ECG) or biosensor data being sent to the monitoring system with minimum power usage by the wearable devices. With the dynamic security scaling system, the cryptographic complexity of usage will be adjusted depending on the sensitivity of the data and the level of network congestion, balancing security and

energy efficiency. The method can include anomaly detection and access control based on blockchain to avert unauthorized access and cyber threats in IIoT applications whereby WSNs monitor the machineries, environmental conditions, and predictive maintenance. Furthermore, with hierarchical clustering and edge computing, this technique will cope with scalability and achieve smart massive sensor data processing. By implementing adaptive machine-learning modelling for anomaly detection and security enhancement, the system can be efficiently designed according to the specific energy requirements, latency requirements, and security needs of different WSN applications.

An adaptive security model can be modeled whereby encryption complexity can dynamically vary based on prevailing network conditions and the level of threat, thus optimizing security and conserving energy in WSN. This type of model can provide for the real-time monitoring of node energy levels, traffic patterns, and attack probabilities, using all these parameters to adjust cryptographic systems—lightweight encryption under normal conditions, whereas a heavier one is used under scenarios of attack. However, large-scale implementation of AROA-based parameter optimization in WSNs poses several challenges involving higher computational overhead, longer convergence time, and limitations on the scale of operation when used to model thousands of sensor nodes. Furthermore, a machine-learning-based anomaly detection model allows a more proactive approach to security wherein it identifies suspected attacks before they can inflict any damage; this thus offers a better long-term solution in comparison to static forms of cryptography such as OFECS. A downside of the ML approach is that, except for a cryptographic enhancement that provides immediate and deterministic security, such algorithms require constant training, large processing resources, and well-designed datasets. Therefore, a better solution could be the hybridization of ML-based anomaly detection to proactively detect possible threats with adaptive cryptographic mechanisms for the protection of real-time data in energy-constrained WSNs.

5. Conclusion

In this research, a new model is proposed, GLCR-OFECS, to provide safe and energy-efficient transmission of data in WSNs. The GLCR was applied to choose the most appropriate forwarder nodes depending on residual energy, packet loss, and link quality. Also, the OFECS was used to enhance data security via encryption and digital signature schemes. AROA combination also maximizes cryptography parameters, which increases randomness, strength, and efficiency in key generation. This optimization enables trading off the security strength and the EC, and the proposed framework is applicable in real-time WSN applications. The effectiveness of GLCR-OFECS is confirmed by extensive simulation results that show that it performs better with a 99.3% PDR, 0.8 mJ EC, 1.3 ms EED, 245 kbps throughput and 512 sec network lifetime for 100 nodes compared to the current methods. Equally, it has 98.57% data confidentiality, 97.74% data integrity, 26.1 MB space complexity, and 32 ms execution time on 2000 data versus current schemes in WSN to transfer data. Nevertheless, challenges like scaling in highly dynamic and dense networks have not been studied. Future work will involve integrating new methods like blockchain-based authentication, and machine learning-based anomaly detection to further enhance scalability, robustness, and resilience of complex WSNs.

References

[1] R. Dogra, S. Rani, Kavita, J. Shafi, S. Kim, M.F. Ijaz, ESEERP: Enhanced Smart Energy efficient Routing Protocol for Internet of Things in Wireless Sensor Nodes. Sensors, 22(16), (2022) 6109. https://doi.org/10.3390/s22166109

[2] M. Rami Reddy, M.L. Ravi Chandra, P. Venkatramana, R. Dilli, Energy-Efficient Cluster head selection in Wireless Sensor Networks using an improved Grey Wolf Optimization Algorithm. Computers, 12(2), (2023) 35. https://doi.org/10.3390/computers12020035

[3] G.C. Jagan, P. Jesu Jayarin, Wireless Sensor Network Cluster Head Selection and Short Routing using energy Efficient Electrostatic Discharge Algorithm. Journal of Engineering, (2022) 1–10. https://doi.org/10.1155/2022/8429285

[4] R. Abraham, M. Vadivel, An Energy Efficient wireless sensor network with Flamingo Search Algorithm-based Cluster Head Selection. Wireless Personal Communications, 130(3), (2023)1503–1525. https://doi.org/10.1007/s11277-023-10342-2

[5] H. Mohapatra, A.K. Rath, R.K. Lenka, R.K. Nayak, R. Tripathy, Topological Localization Approach for Efficient Energy Management of WSN. Evolutionary Intelligence, 17(2), (2024) 717–727. https://doi.org/10.1007/s12065-021-00611-z

[6] S. Hao, Y. Hong, Y. He, An Energy-Efficient Routing Algorithm based on Greedy Strategy for Energy Harvesting Wireless Sensor Networks. Sensors, 22(4) (2022) 1645. https://doi.org/10.3390/s22041645

[7] S. Madhu, R.K. Prasad, P. Ramotra, D.R. Edla, A. Lipare, A Location-Less Energy Efficient Algorithm for load Balanced Clustering in Wireless Sensor Networks. Wireless Personal Communications, 122(2), (2022) 1967–1985. https://doi.org/10.1007/s11277-021-08976-1

[8] P. Kathiroli, K. Selvadurai, Energy Efficient Cluster Head Selection using Improved Sparrow

Search Algorithm in Wireless Sensor Networks. Journal of King Saud University – Computer and Information Sciences, 34(10) (2022) 8564–8575. https://doi.org/10.1016/j.jksuci.2021.08.031

[9] S. Urooj, S. Lata, S. Ahmad, S. Mehfuz, S. Kalathil, Cryptographic Data Security for Reliable Wireless Sensor Network. Alexandria Engineering Journal, 72, (2023) 37–50. https://doi.org/10.1016/j.aej.2023.03.061

[10] D. Gammelli, K.P. Rolsted, D. Pacino, F. Rodrigues, Generalized Multi-Output Gaussian Process Censored Regression. Pattern Recognition, 129, (2022) 108751. https://doi.org/10.1016/j.patcog.2022.108751

[11] M. Kumar, P. Mukherjee, K. Verma, S. Verma, D.B. Rawat, Improved Deep Convolutional Neural Network based Malicious Node Detection and Energy-Efficient Data Transmission in Wireless Sensor Networks. IEEE Transactions on Network Science and Engineering, IEEE, 9(5), (2021) 3272–3281. https://doi.org/10.1109/TNSE.2021.3098011

[12] S.S. Kalburgi, M. Manimozhi, Taylor-Spotted Hyena Optimization Algorithm for Reliable and Energy-Efficient Cluster Head Selection based Secure Data routing and Failure Tolerance in WSN. Multimedia Tools and Applications, 81(11), (2022) 15815–15839. https://doi.org/10.1007/s11042-022-12302-7

[13] S. Vijayalakshmi, G. Kavithaa, N.V. Kousik, Improving Data Communication of Wireless Sensor Network using Energy Efficient Adaptive Cluster-Head Selection Algorithm for secure routing. Wireless Personal Communications, 128(1), (2023) 25–42. https://doi.org/10.1007/s11277-021-09398-9

[14] S.G. Qureshi, S.K. Shandilya, Novel fuzzy based Crow Search Optimization Algorithm for Secure node-to-node Data Transmission in WSN. Wireless Personal Communications, 127(1), (2022) 577. https://doi.org/10.1007/s11277-021-08352-z

[15] G. Arya, A. Bagwari, D.S. Chauhan, Performance Analysis of Deep Learning-based Routing Protocol for an efficient data transmission in 5G WSN communication. IEEE Access, 10, (2022) 9340–9356. https://doi.org/10.1109/ACCESS.2022.3142082

[16] L. Hu, C. Han, X. Wang, H. Zhu, J. Ouyang, Security Enhancement for Deep Reinforcement Learning-based Strategy in Energy-Efficient Wireless Sensor Networks. Sensors, 24(6), (2024) 1993. https://doi.org/10.3390/s24061993

[17] A. Sharma, A. Kansal, Advanced ANN based secured energy efficient routing protocol in WSN. Wireless Personal Communications, 132(4), (2023) 2645–2666. https://doi.org/10.1007/s11277-023-10737-1

[18] A. Jalili, J.A. Alzubi, R. Rezaei, J.L. Webber, C. Fernández-Campusano, M. Gheisari, R. Amin, A. Mehbodniya, Markov Chain-based Analysis and Fault Tolerance Technique for Enhancing Chain-based Routing in WSNs. Concurrency and Computation: Practice and Experience, 36(12), (2024) e8032. https://doi.org/10.1002/cpe.8032

[19] A. Alrabea, O.A. Alzubi, J.A. Alzubi, A Task-Based Model for Minimizing Energy Consumption in WSNs. Energy Systems, 13, (2022) 671–688. https://doi.org/10.1007/s12667-019-00372-w

[20] A. Jalili, M. Gheisari, J.A. Alzubi, C. Fernández-Campusano, F. Kamalov, S. Moussa, A Novel Model for Efficient Cluster Head Selection in Mobile WSNs using Residual Energy and Neural Networks. Measurement: Sensors, 33, (2024) 101144. https://doi.org/10.1016/j.measen.2024.101144

[21] I. Qiqieh, J.A. Alzubi, O.A. Alzubi, DNA Cryptography based Security Framework for Health Cloud Data. Computing, 107(1), (2024) 35. https://doi.org/10.1007/s00607-024-01393-9

[22] O.A. Alzubi, J.A. Alzubi, O. Dorgham, M. Alsayyed, Cryptosystem Design based on Hermitian Curves for IoT Security. Journal of Supercomputing, 76, (2020) 8566–8589. https://doi.org/10.1007/s11227-020-03144-x

[23] G. Sakthivel, P. Madhubala, Advanced Set Containment Deep Learned Rabin Certificateless Signcryption for Secured Transmission with Big Data in Cloud. Concurrency and Computation: Practice and Experience, 36(1), (2024) e7883. https://doi.org/10.1002/cpe.7883

[24] S. Kumar, D. Sharma, Key Generation in Cryptography using Elliptic-Curve Cryptography and Genetic Algorithm. Engineering Proceedings, 59(1), (2023) 59. https://doi.org/10.3390/engproc2023059059

[25] S. Nagaraj, A.B. Kathole, L. Arya, N. Tyagi, S.B. Goyal, A.S. Rajawat, M.S. Raboaca, T.C. Mihaltan, C. Verma, G. Suciu, Improved Secure Encryption with Energy Optimization using Random Permutation Pseudo Algorithm based on Internet of Things in Wireless Sensor Networks. Energies, 16(1), (2022) 8. https://doi.org/10.3390/en16010008

[26] O.A. Khashan, N.M. Khafajah, W. Alomoush, M. Alshinwan, Innovative Energy-Efficient Proxy Re-Encryption for Secure Data Exchange in Wireless Sensor Networks. IEEE Access, IEEE, 12, (2024) 23290-23304. https://doi.org/10.1109/ACCESS.2024.3360488

Authors Contribution Statement

J. Paruvathavardhini: Conceptualization, Methodology, Software, Writing-Original draft, Visualization, Writing - Review & Editing. B. Sargunam: Investigation, Supervision, Writing - Review & Editing. Both authors

have read and agreed to the published version of the manuscript.

Funding

The authors declare that no funds, grants or any other support were received during the preparation of this manuscript.

Competing Interests

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Data Availability

The data supporting the findings of this study can be obtained from the corresponding author upon reasonable request.

Has this article screened for similarity?

Yes