SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V11I12P117

Volume 11 Issue 12, 171-188, December 2024

Original Article

Design of Effective Seeker Optimized Quantum Lightweight Cryptography Using Hybrid Redundant Quantum Key Distribution

K.U.V. Padma[1], E. Neelima[2]

[1,2]*Department of Computer Science and Engineering, GITAM (Deemed to be University), Andhra Pradesh, India.*

[1]*Corresponding Author : kuvpadma@gmail.com*

Received: 22 October 2024 Revised: 26 November 2024 Accepted: 13 December 2024 Published: 30 December 2024

Abstract *- Ultra Lightweight Cryptography (ULWC) is a critical field of research with a primary focus on developing cryptographic algorithms tailored for resource-constrained devices and applications. The challenge in such environments is to ensure robust security and efficient performance, as conventional security protocols often fail to meet these dual objectives. To address these issues, this work explores a hybrid approach, specifically Optimized Quantum Lightweight Cryptography (OQLC), which guarantees secure data transmission and protection in resource-constrained settings. The heart of OQLC lies in the fusion of the ULWC-based Efficient Randomized-Grain (ERG)-128 algorithm and the Redundant Quantum Key Distribution (RQKD-QC) framework. An Effective Seeker Optimization (ESO) technique is employed to maximise the performance of these two algorithms. ESO harnesses the inherent parallelism and adaptability of natural seekers to optimize various parameters of the OQLC algorithm, including key scheduling, round functions, and other cryptographic primitives. Integrating ESO with the hybrid ERG-128 and RQKD-QC algorithm enhances the overall performance and efficiency of the cryptographic system. This equilibrium balances lightweight implementation, post-quantum security, and improved performance, thereby addressing the unique demands of resource-constrained environments. In a comparative analysis of encryption and decryption times, the proposed OQLC method demonstrates impressive efficiency. It accomplishes encryption in a mere 0.00101 seconds, while decryption is achieved in a mere 0.00023 seconds.*

Keywords *- Effective seeker optimization, Quantum cryptography, Redundant quantum key distribution, Ultra lightweight cryptography, Optimized quantum lightweight cryptography.*

1. Introduction

ULWC is an area of specialized research that focuses on creating cryptographic algorithms and protocols that are intended to be very effective and lightweight in terms of the amount of computing resources, the amount of memory used, and the amount of energy used [1]. The growing necessity for protected communication and data storage in applications and devices with limited resources prompted the development of ULWC to solve this problem. In recent years, there has been a fast development of devices connected to the Internet of Things (IoT), embedded systems, and other tiny computer devices [2].

These devices have limited processing capacity, memory, and battery life. These devices are used in various domains, such as healthcare, automotive, industrial automation, and smart homes, where security and privacy are paramount. However, traditional cryptographic algorithms [3], primarily designed for high-end systems, often prove impractical or inefficient when implemented on these lightweight devices. Many devices have limited computational resources, including low-power microcontrollers, sensors, and actuators [4]. These devices are typically characterized by their low processing power, limited memory, and constrained energy supply. Implementing traditional cryptographic algorithms on such devices would impose a significant computational burden and quickly drain the battery. ULWC algorithms are designed to be lightweight and demand very few computing resources, making them ideal for devices with limited storage space. Real-time monitoring is important in certain applications, such as wireless sensor networks [5]. In control systems, there are strict limitations on bandwidth and latency.

Traditional cryptographic algorithms tend to generate large ciphertexts and involve complex operations, which can lead to increased communication overhead and significant delays. ULWC algorithms [6] are designed to produce compact ciphertexts and minimize computational

complexity, enabling efficient communication and reducing latency. The need for cost-effective security solutions also drives the ULWC. Many IoT devices are mass-produced and cost-sensitive, and integrating expensive hardware or implementing complex cryptographic algorithms is not economically viable [7]. ULWC algorithms can be implemented with minimal hardware requirements and are designed to be computationally efficient, enabling manufacturers to provide cost-effective security solutions without compromising the level of protection. The increasing connectivity of devices and the widespread adoption of IoT technologies have exposed numerous security threats and vulnerabilities [8].

Lightweight devices are particularly vulnerable to attacks due to their limited resources and constrained security measures. ULWC algorithms address these vulnerabilities by providing robust security features tailored for lightweight devices [9]. They aim to provide a high level of security while mitigating the risks associated with limited resources. With the rise of data-driven applications and services, ensuring privacy and data protection has become crucial. Many lightweight devices handle sensitive personal information and require secure communication to prevent unauthorized access and data breaches. ULWC algorithms provide cryptographic primitives, such as encryption and authentication, that are tailored for lightweight devices, ensuring the confidentiality and integrity of data even in resource-constrained environments [10]. Various industries and sectors have specific regulatory and compliance requirements for security and privacy. ULWC algorithms play a vital role in meeting these requirements by providing lightweight cryptographic solutions that adhere to industry standards and guidelines. They enable organizations to implement secure systems and applications in compliance with regulations without compromising performance or cost.

So, ULWC must address the unique challenges that resource-constrained devices and applications pose. Despite limited resources, it provides efficient cryptographic algorithms and protocols specifically designed for lightweight devices, ensuring secure communication, data protection, and privacy. ULWC enables the widespread adoption of IoT technologies, embedded systems, and other lightweight devices while maintaining high security, cost-effectiveness, and compliance with regulatory requirements. The novel The OQLC combines two powerful encryption techniques, ULWC and post-quantum cryptography. The ERG-128-based ULWC approach is specifically designed for resource-constrained environments, providing strong security and lightweight implementation. The RQKD-QC framework helps to protect sensitive data from potential quantum attacks, which ensures that even with advances in quantum computing, the encrypted data remains secure. The work introduces an ESO optimisation technique, which takes advantage of seekers' natural parallelism and adaptability. ESO optimizes different aspects of the OQLC algorithm, such as key scheduling and round functions, resulting in improved performance without compromising security. The OQLC algorithm strikes a balance between being lightweight and efficient. It offers strong protection for sensitive data while being suitable for resource-constrained environments. The remaining parts of the paper are structured as follows: section 2 focused on the literature survey of various ULWC methods. Further, section 3 focused on the OQLC framework with ERG-128 based ULWC, RQKD-QC based quantum cryptography, and ESO models. Further, section 4 focused on the simulation results of the OQLC framework. Finally, the essay ends with a discussion of potential future scope in section 5.

2. Literature Survey

El Hadj Youssef and colleagues [11] suggested a revised model of the LEON3 CPU that they referred to as the ReonV Reduced Instruction Set Computer-Five (RISCV) processor. This variant of LEON3 was specifically designed for IoT applications, incorporating robust and effective security features from the initial stages of its design process. Liu et al. [12] advocated enhancing the encryption and decryption capabilities of the lightweight cipher uBlock to improve overall performance and safety in IoT connectivity. The proposed solution achieves the necessary degree of security and communication performance while minimizing energy consumption. Panchami et al. [13] proposed using the Feather S-box, a 4-bit, highly nonlinear, bijective, and balanced S-box to create confusion in lightweight cyphers. The authors analysed the Feather S-box's hardware performance in-depth, considering factors such as area and crucial path-delay cost. El-Hajj et al. [14] focused on lightweight symmetric ciphers for resource-constrained devices. They evaluated 39 alternative block ciphers by implementing them on an ATMEGA328p microcontroller, analysing their speed, cost, and energy efficiency in encrypting and decrypting data with varying block sizes and key lengths.

Prakasam et al. [15] proposed the Hybrid Lightweight Cryptography Authentication Scheme (HLCAS), prioritising low latency, minimal area usage, and maximum power efficiency. The scheme utilizes the concept of 8-bit manipulation and demonstrates the challenges of hardware implementation using FPGA devices of the Spartan3E XC3S500E kind. Windarta et al. [16] proposed a few lightweight cryptographic hash function strategies for constrained systems. The authors analyzed and classified cryptanalytic assaults, cryptographic characteristics and design trends, compared various hardware and software implementations, and proposed new lightweight cryptographic hash functions. An innovative, secure end-to-end Internet of Things communication system was proposed by Winarno et al. [17]. The technique used lightweight cryptography that was based on a block cipher. The protocol

provides a safe basis for communication, which uses the Galantucci technique for secret exchange in conjunction with a lightweight cryptographic algorithm. Alshehri et al. [18] created an Attribute-based Access Control approach for the Internet of Things (AAC-IoT) by using the blockchain technology provided by Hyperledger Fabric (HLF). This approach overcomes security issues by requiring data owners to register and verify themselves within the AAC-IoT system, leveraging identities, certificates, and signatures. Kurniawan et al. [19] presented a novel low-overhead secure communication protocol developed on the Arduino platform. The SPECK lightweight block cipher, the BLAKE2s hash function, and a lightweight key agreement method are all included within the protocol. In the Internet of Things (IoT) context, Jammula et al. [20] offered the LWC-ABE strategy to improve security performance against various threats. The technique that has been presented cuts down on the number of trusted authority environments, which eliminates a potential bottleneck in Internet of Things servers and devices. Lightweight cryptographic methods were presented as a solution for wireless Internet of Things networks in Blanc et al.'s [21] study.

The authors ported all 12 algorithms to various hardware platforms and evaluated their performance on various platforms, including x86_64 PC, MSP430F1611, AVR ATmega128, and the IoT-LAB platform. An architecture for lightweight cryptographic primitives was proposed by Tsantikidou et al. [22], who also studied the constraints of these primitives in terms of general hardware, security, and architectural considerations. The efficiency of the algorithms was measured and compared based on how well they protected healthcare applications, the device used, and the overall implementation efficiency. Salem et al. [23] proposed a lightweight encryption/decryption approach for the IoT, specifically in the context of biosensors and bio-actuators. Despite the disorganized nature of the data, the suggested encryption approach provides efficient and secure protection for sensitive medical information. Goyal et al. [24] assessed security algorithms, comparing their performance and robustness.

The authors suggested several energy-efficient and lightweight cryptographic techniques suitable for IoT devices. These comparisons were conducted through hardware implementation and cryptographic analysis." The researchers Ahmed et al. [25] proposed an unforgeable digital signature that could be put into an Efficient Lightweight Encryption (ELCD) technique. This technique fixes the weak bit issue brought on by the Diffie-Hellman exchange. It does this via secure key distribution in Elliptic Curve Diffie-Hellman (ECDH). The ELCD approach combines digital signature with encryption, using rapid hash functions that enable the private transit of shared secret keys across IoT devices. This was accomplished even over insecure communication channels. Mhaibes et al. [26] suggested modifying TEA by creating a new key generation function that used two Linear Feedback Shift Registers (LFSRs) to solve security issues caused by using separate keys in each round function. Lightweight Cryptography was addressed in the paper by Im et al. [27], which presented the S-Box Attack Using FPGA Reverse Engineering as a potential solution. Using this method, it is possible to successfully extract the 64-bit plaintext for ciphers such as DESL, LBlock, and TWINE.

In the case of KLEIN and LED, all 64-bit keys have been completely recovered. However, in the case of PRESENT, only 80 percent of the 64-bit keys (out of a total of 80-bit keys) have had some of their bits returned. Mohammad et al. [28] presented an Advanced Encryption Standard (AES) method that reduces the required computing power and enhances cryptography performance for devices with limited resources. Goulart et al. [29] discussed lightweight encryption techniques for the IoT, focusing on making well-known ciphers like AES and Elliptic Curve Cryptography (ECC) more lightweight. Gupta et al. [30] suggested lightweight cryptography algorithms and protocols for low-power IoT devices. The authors emphasized various international standards organizations that facilitate the rapid development of IoT-enabled protocols.

2.1. Research Gaps

The research in lightweight cryptography and its applications for IoT presents several gaps. First, while many studies focus on the development of lightweight cryptographic algorithms and their efficiency, there is a need for more research into real-world implementation challenges, especially in resource constrained IoT devices. This includes investigations into the trade-offs between security, performance, and energy consumption in practical IoT scenarios. Second, there is a lack of standardization in lightweight cryptography for IoT, making it challenging for developers to choose suitable cryptographic solutions for their specific applications. Establishing industry standards and best practices could help bridge this gap. Additionally, the interoperability of various lightweight cryptographic methods across IoT platforms and devices remains a significant challenge. Finally, as the IoT ecosystem grows, there is a growing concern about scalability and the ability of lightweight cryptographic techniques to adapt to the increasing complexity and diversity of IoT networks. Addressing these research gaps is essential for the continued development and security of IoT systems.

3. Proposed Optimized Quantum Lightweight Cryptography

The work focuses on developing a hybrid approach called OQLC for secure transmission and protection of sensitive data in resource-constrained environments. Figure 1 shows the OQLC block diagram, which combines two

cryptographic algorithms [31]. ERG-128, a ULWC algorithm and post-quantum cryptographic methods based on the RQKD-QC framework are here. Additionally, an ESO technique is employed to optimize the performance of these algorithms. The primary goal of OQLC is to balance lightweight implementation, post-quantum security, and improved performance. By combining the lightweight ERG-128 algorithm with post-quantum cryptographic methods, the system aims to provide enhanced security while maintaining efficiency in resource-constrained environments. The RQKD-QC framework incorporates post-quantum security into the OQLC algorithm [32]. Post-quantum cryptography is intended to resist assaults by quantum computers, which can break standard cryptographic techniques. Its design is based on the idea that quantum computers would become more common. OQLC can provide better protection against potential vulnerabilities posed by future quantum computing developments by incorporating post-quantum cryptographic algorithms [33].

The ESO technique is employed to optimize the performance of OQLC. ESO leverages natural seekers' inherent parallelism and adaptability to optimize various parameters of the OQLC algorithm, including key scheduling, round functions, and other cryptographic primitives. This optimization technique aims to improve the overall performance and efficiency of the cryptographic system [34]. By integrating ESO with the hybrid ERG-128 and RQKD-QC algorithm, the OQLC approach enhances the performance and efficiency of the cryptographic system. It addresses the limitations of conventional security protocols by providing a lightweight implementation suitable for resource-constrained devices and applications. Moreover, it incorporates post-quantum security to protect against potential quantum computing attacks, making it a robust solution for securing sensitive data.

3.1. Efficient Randomized-Grain-128 Based ULWC

The ERG-128 algorithm is a stream cipher belonging to the family of ULWC encryption algorithms. Its purpose is to create a series of pseudorandom bits, known as a keystream, which yield the desired result when coupled with the plaintext to form ciphertext or with ciphertext to produce plaintext for encryption or decryption [35]. The ERG-128 is constructed with a non-linear feedback function and a linear feedback shift register (also known as an LFSR). Figure 2 shows the ERG-128 based ULWC flowchart, and Table 1 shows the algorithm of ERG-128 based ULWC. To provide a detailed mathematical analysis of ERG-128, its key components and operations are illustrated as follows:

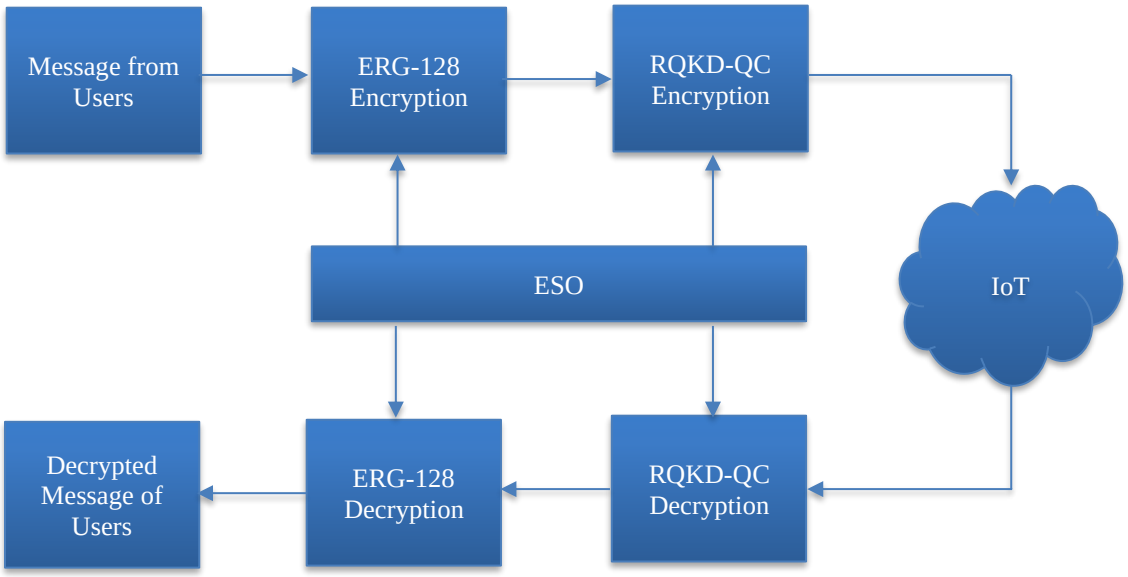

Fig. 1 Proposed optimized quantum lightweight cryptography operational diagram

3.2. Linear Feedback Shift Register (LFSR)

The LFSR is a shift register that consists of a chain of flip-flops with feedback connections that introduce linear feedback. In ERG-128, the LFSR has a length of 128 bits, denoted by $LFSR\,[0], LFSR\,[1], \ldots, LFSR[127]$.

The LFSR is updated at each clock cycle using the following update function:

$$A=LFSR\,[(i-1) \bmod 128] \oplus LFSR[(i-61) \bmod 128] \quad (1)$$

$$B= (LFSR[(i-101) \bmod 128] \,\&\, LFSR[(i-128) \bmod 128]) \quad (2)$$

$$LFSR[i] = A \oplus B \quad (3)$$

Here, "$\oplus$" denotes the bitwise exclusive OR (XOR) operation, and "&" represents the bitwise AND operation.

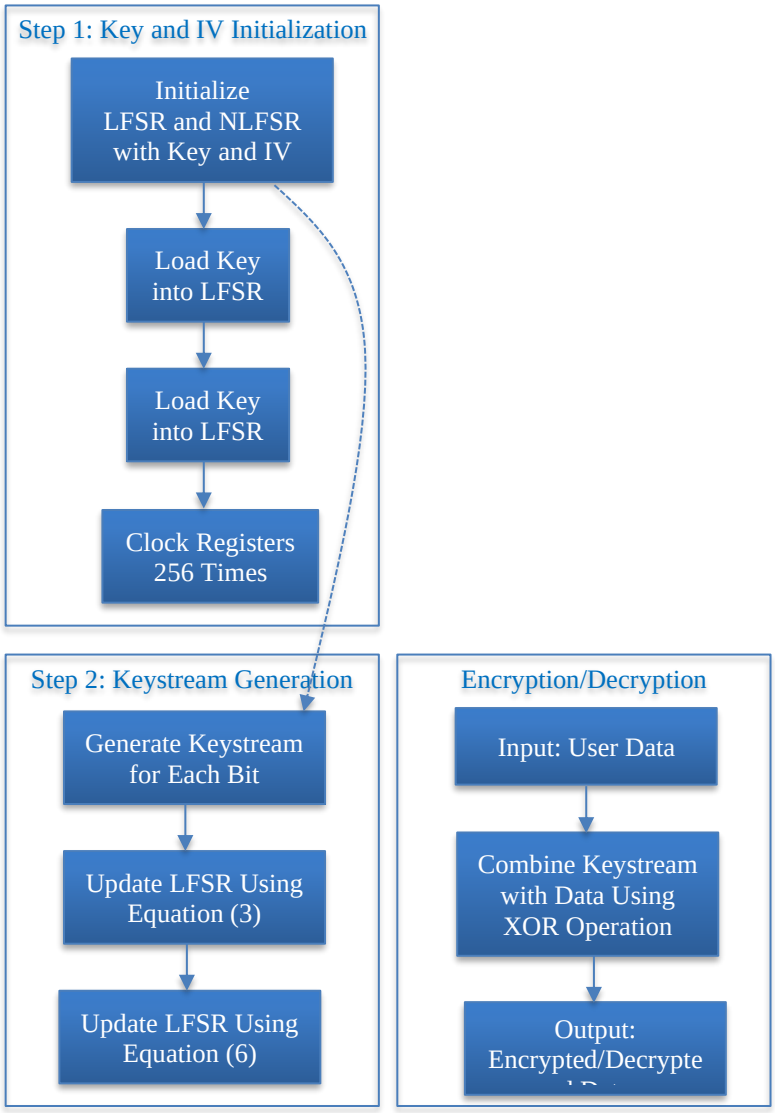

Fig. 2 Efficient randomized-grain-128 based ULWC flowchart

3.3. Non-Linear Feedback Function

The ERG-128's non-linear feedback function combines the output of the LFSR with a non-linear function that offers extra diffusion and security [36]. The combination of the LFSR output and the non-linear function does this. The function will be defined in the following manner:

$$C = LFSR[(i - 1)\ mod\ 128] \quad (4)$$

$$D = (LFSR[(i - 26)\ mod\ 128] \,\&\, LFSR[(i - 70)\ mod\ 128]) \quad (5)$$

$$NLFSR[i] = C \oplus D \oplus LFSR[(i - 91) mod\ 128] \quad (6)$$

The NLFSR output is then fed into the LFSR as part of the update function.

3.3.1. Initialization

ERG-128 requires a 128-bit secret key and a 96-bit initialization vector (IV) as inputs. The key and IV are used to initialize the LFSR and NLFSR registers before generating the keystream.

3.3.2. Keystream Generation

Once the LFSR and NLFSR registers are initialized, the keystream generation proceeds by repeatedly updating the registers and producing the output bit. The keystream is generated as follows:

$$Keystream[i] = NLFSR[0] \oplus LFSR[0] \quad (7)$$

The NLFSR and LFSR are updated simultaneously at each clock cycle.

Table 1. Algorithm of ERG-128 based ULWC

Input: User data Output: ERG-128 Encrypted outcome
Step 1: Key and IV Initialization: Input: 128-bit secret key (K) and 96-bit initialization vector (IV). Initialize the LFSR and NLFSR registers with the key and IV values. Load the key into the LFSR and the IV into the NLFSR. Clock the registers 256 times to ensure proper initialization. Step 2: Keystream Generation: Repeat the following steps to generate each keystream bit. Calculate the output bit using Equation (7). Update the LFSR using Equation (3). Update the NLFSR using Equation (6). Step 3: Encryption/Decryption: To encrypt or decrypt data using ERG-128, combine the generated keystream with the plaintext or ciphertext using the bitwise XOR operation. Each keystream bit is XORed with the corresponding bit in the plaintext or ciphertext to produce the resulting ciphertext or plaintext, respectively.

3.3.3. Encryption / Decryption

To encrypt or decrypt data using ERG-128, the keystream generated from the algorithm is combined with the plaintext or ciphertext using the bitwise XOR operation. It is necessary to perform an XOR operation on each keystream bit with its corresponding bit in the plaintext or ciphertext to generate the ciphertext or plaintext, respectively [37]. Mathematical analysis of ERG-128 involves studying the properties and behavior of the LFSR, NLFSR, and the keystream generation process. This analysis typically includes assessing the algorithm's resistance against known attacks, evaluating its statistical properties (such as randomness and correlation), and verifying its security guarantees, such as the avalanche effect and diffusion.

3.4. Redundant Quantum Key Distribution Based Quantum Cryptography

RQKD-QC is a cryptographic protocol that utilizes quantum mechanics to securely exchange encryption keys between two parties. In RQKD-QC, the transmission of quantum states is redundant, meaning multiple copies of the quantum state are sent to mitigate the effects of noise and eavesdropping attacks [38]. Figure 3 shows the RQKD-QC flowchart, and Table 2 shows the algorithm of RQKD-QC. In quantum mechanics, Quantum states are used to store information; usually, these states are represented by vectors in a complex vector space known as a Hilbert space. The qubit is the fundamental unit of information, and it can be in a superposition of two states, traditionally represented by the notation 0 and 1.

These two states were represented, respectively, by the column vectors [1, 0] and [0, 1]. Operators describe measurements on qubits as projectors. A measurement on a standard basis (computational basis) projects the qubit onto one of the basis states, yielding either 0 or 1 as the outcome [39]. Establishing a shared secret key between two parties, which are conventionally known as sender and receiver, is the primary objective of quantum key distribution and receiver while ensuring its confidentiality and detecting eavesdropping attempts. The RQKD-QC protocol steps are defined as follows:

3.4.1. Step 1

Preparation Phase: In this phase, the sender prepares a sequence of qubits in a random state. Let us denote the state of each qubit prepared by the sender $|\psi\rangle$. This was written as a linear combination of zero and one basis states.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (8)$$

Here, α and β are complex probability amplitudes satisfying the normalization condition $|\alpha|^2 + |\beta|^2 = 1$. Using various quantum states, the sender randomly encodes bits (0s and 1s) onto the qubits. For example, she chooses to encode a 0 bit as $|\psi_0\rangle$ and a 1 bit as $|\psi_1\rangle$.

3.4.2. Step 2

Transmission Phase: The sender sends multiple copies of each qubit to the receiver over a quantum channel. Let us say the sender sends N copies of each qubit to the receiver. The state of the transmitted qubits can be represented as:

$$|\psi_{transmitted}\rangle = |\psi_0\rangle \otimes |\psi_0\rangle \otimes \ldots \otimes |\psi_0\rangle + |\psi_1\rangle \otimes |\psi_1\rangle \otimes \ldots \otimes |\psi_1\rangle \quad (9)$$

Where, $\otimes$ represents the tensor product.

3.4.3. Step 3

Measurement Phase: The receiver randomly chooses a measurement basis for each received qubit. Let us denote the measurement basis for the i^{th} qubit as $\{|0\rangle\_i, |1\rangle\_i\}$, where i ranges from 1 to N. The Receiver performs measurements on each qubit and records the outcomes. The measurement outcome for the i^{th} qubit can be denoted as y_i, where $y_i = 0 \; or \; y_i = 1$.

3.4.4. Step 4

Information reconciliation: sender and receiver compare a subset of their measurement outcomes to estimate the error rate introduced by noise and eavesdropping. Let us assume they choose a subset of M measurement outcomes for comparison. To estimate the error rate, they compute the error rate ε as:

$$\varepsilon = \frac{(M - \Sigma_i y_i \oplus x_i)}{M} \quad (10)$$

Where, Σ denotes summation, y_i is the measurement outcome of the receiver, x_i is the bit value encoded by the sender and $\oplus$ represents bitwise XOR operation. They use

error correction techniques, such as error correction codes like the binary symmetric channel model, to reconcile their data and obtain an agreed-upon subset of matching bits [40].

3.4.5. Step 5

Privacy Amplification: The sender and receiver apply privacy amplification algorithms to extract a shorter, secure key from their matching bits. The goal is to eliminate any potential information an eavesdropper possesses. Privacy amplification typically involves hashing algorithms and information-theoretic tools. Let us denote the matching bits as z_i. The sender and receiver can derive a secure key K of length L from the matching bits using a privacy amplification function f:

$$K = f(z^1, z^2, \ldots, z_M) \quad (11)$$

The privacy amplification function ensures that even if an eavesdropper has partial knowledge of the matching bits, the resulting key is secure and independent of the eavesdropper's information.

Table 2. Algorithm of RQKD-QC

Input: ERG-128 Encrypted Data Output: RQKD-QC Encrypted outcome
Step 1: Preparation Phase Step 1.1: Sender uses a single-photon source to prepare a sequence of qubits in random states. Step 1.2: The sender randomly assigns bits (0s and 1s) to the qubits using different quantum states. Step 2: Transmission Phase: The sender sends multiple copies of each qubit to the Receiver through a quantum channel. Step 3: Measurement Phase Step 3.1: The receiver randomly selects a measurement basis for each received qubit. Step 3.2: The receiver measures each qubit and records the outcomes. Step 4: Information Reconciliation Step 4.1: Sender and Receiver compare a subset of their measurement outcomes to estimate the error rate caused by noise and eavesdropping. Step 4.2: They use error correction techniques to reconcile their data and obtain a set of matching bits. Step 5: Privacy Amplification Step 5.1: Sender and Receiver apply privacy amplification algorithms to derive a shorter, secure key from the matching bits. Step 5.2: This ensures the final key's security, even if an eavesdropper possesses some information.

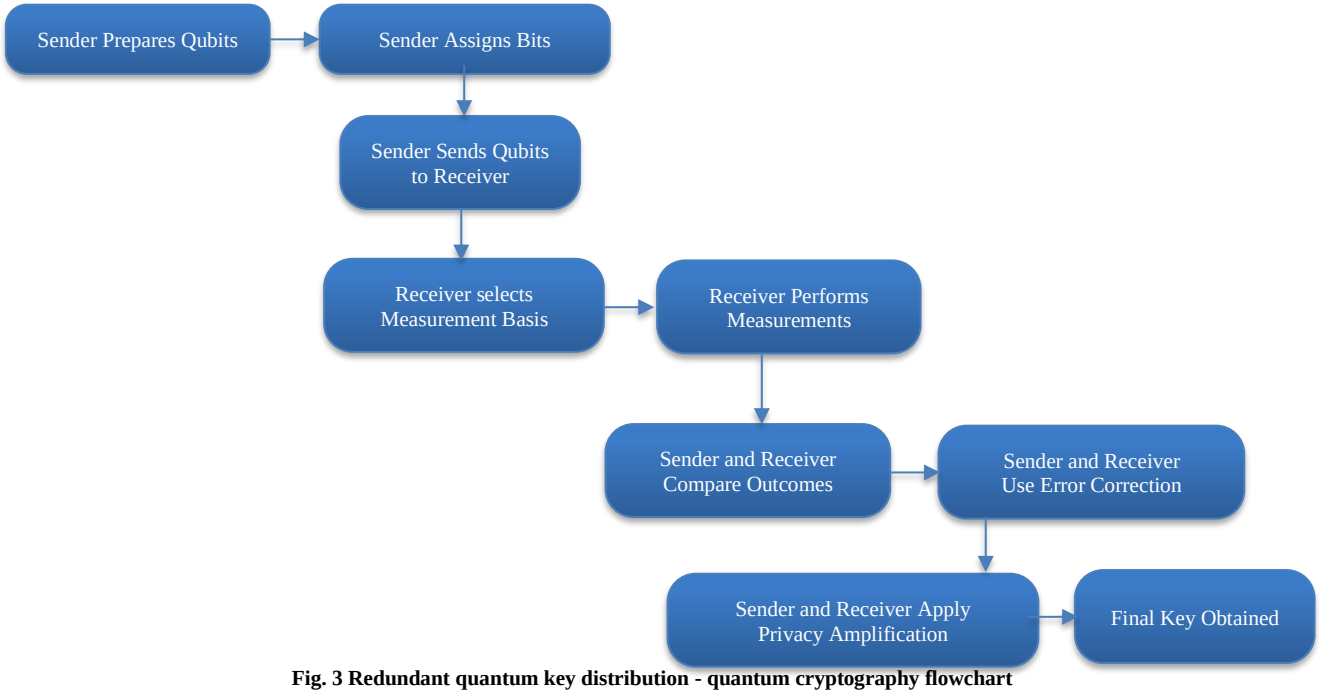

Fig. 3 Redundant quantum key distribution - quantum cryptography flowchart

3.5. Effective Seeker Optimization Algorithm

Figure 4 shows the ESO optimization flowchart, and Table 3 shows the algorithm of ESO optimization. The ESO is modelled after the way humans look for information. The whole population of people looking for jobs is broken up into three subpopulations of almost equivalent size. $P_i; j_i$ 14 1; 2;…; S_N; j_i 14 1; 2;…D is the name of the two-dimensional matrix representing the population of candidate solutions. Here, S_N stands for the size of the population, and D speaks for the size of the issue (the number of problem variables). The corresponding vector x_i: x_{i1}; x_{i2};…; x_{iD} in generation t, denoted $x_i(t)$for each seeker, i has the following characteristics: the vector x_{i1} represents the values of the variables in the generation before it. The vector denotes the value of the seeker's personal best position discovered to date p_{bi}, whereas the value of the neighbourhoods all-time best position historically is denoted by the vector n_{best}. Each seeker in the population is self-centred and feels that he is, by his judgment, following his best historical position regarding search direction adjustment. In other words, each seeker believes he should act according to his best historical position. In addition, any person who is searching for something is an egoist. Egoistic empirical direction This behaviour is modelled from a vector $(d_{i,ego}(t))$, which is derived as follows:

$$d_{i,ego}(t) = p_{bi} - x_i(t) \tag{12}$$

On the other hand, the seeker is also a social actor and exhibits aspects of behaviour consistent with altruism. To achieve the desired goal, he intends to speak with the other people in his neighbourhood, work together with them, and modify his conduct in response to what he learns from the other people who are looking. The vector $d_{i,alt}$ in one of the forms decides which direction each seeker i in the population will go in terms of their altruistic behavior.

$$d_{i,alt1} = n_{best} - x_i(t) \tag{13}$$

$$d_{i,alt2} = l_{best} - x_i(t) \tag{14}$$

Here, n_{best} represents the best position that the neighbourhood has ever held historically and l_{best} represents the greatest position that the neighborhood has at present. Seekers are also characterized by their proactive conduct. They are geared at achieving a certain objective. In addition, future conduct can be anticipated and directed by looking at past behaviour. This is the direction vector of empirical proactiveness.

$$d_{i,pro}(t) = x_i(t_1) - x_i(t_2) \tag{15}$$

Where are the best and worst places in the set x_i are indicated by the symbols $t_1; t_2$, respectively. In addition, the expression of the search direction for the i^{th} seeker is the stochastic combination of the direction vectors for egoism, altruism, and proactiveness.

$$d_i(t) = sgn[\omega * d_{i,pro}(t) * \emptyset_1 * d_{i,ego}(t) * \emptyset_2 * d_{i,alt}(t)] \tag{16}$$

Here, ω the moment of inertia is denoted, and random integers with values ranging from 0 to 1 are found. The inertia weight is a control parameter that the algorithm uses to conclude about anything. Its purpose is to lessen the local search impact brought on by the progressively $d_{i,ego}(t)$ of the i^{th} seeker and to strike a balance between the global and local exploitation and exploration of resources. This will be accomplished by gradually reducing the number of local seekers. The inertia weight will normally drop linearly from 0.9 to 0.1 during a single process pass. The sgn function is used on the input vector's variables, also known as parameters. The step length is figured out using a computation method that uses fuzzy logic. Following placing the objective function values of each subpopulation in descending order, sequence numbers ranging from 1 to SS are then allocated to them to increase occurrence. In fuzzy reasoning, these are the inputs, denoted by sequence numbers that run the gamut from 1 to SS.

This paves the way for the strategy to be used for a larger variety of optimization problems than it might have been before. The value of the letter S in the acronym denotes the size of the subpopulation to which the seeker belongs. Seeker *I* members have a membership degree of

$$u_i = u_{max} - \frac{SS - I_i}{SS - 1} * (u_{max} - u_{min}) \tag{17}$$

After the population has been sorted by the values of the goal function in decreasing order, the sequence number of the seeker x_i is denoted by I_i. This occurs after the population has been analysed. The notation denotes the maximum membership degree u_{max}, and its value is often around 1.0. The fuzzy system operates according to the logic of the control rule that is broken down as follows: if (the condition portion) then (the action part).

The action component of the phrase is carried out with the help of the bell membership function.

$$u(x) = e^{-x^2/2\hat{}\delta 2} \tag{18}$$

Because just one variable is considered for simplicity, The values of the membership degree that are more than 1.23 but less than 0.0111 are represented by the input variables. Therefore, the value of u_{min} has been adjusted to 0.0111. The formula that must be used to figure out the value of the bell membership function's argument is as follows:

$$\delta = \omega * \left| X_{best} - X_{avg} \right| \tag{19}$$

Here, x_{best} represents the position of the seeker who is regarded as being the most desirable member of the subgroup that the i^{th} seeker belongs, and xavg represents the position that is regarded as being the position that is considered to be the average of all the seekers who belong to the same group.

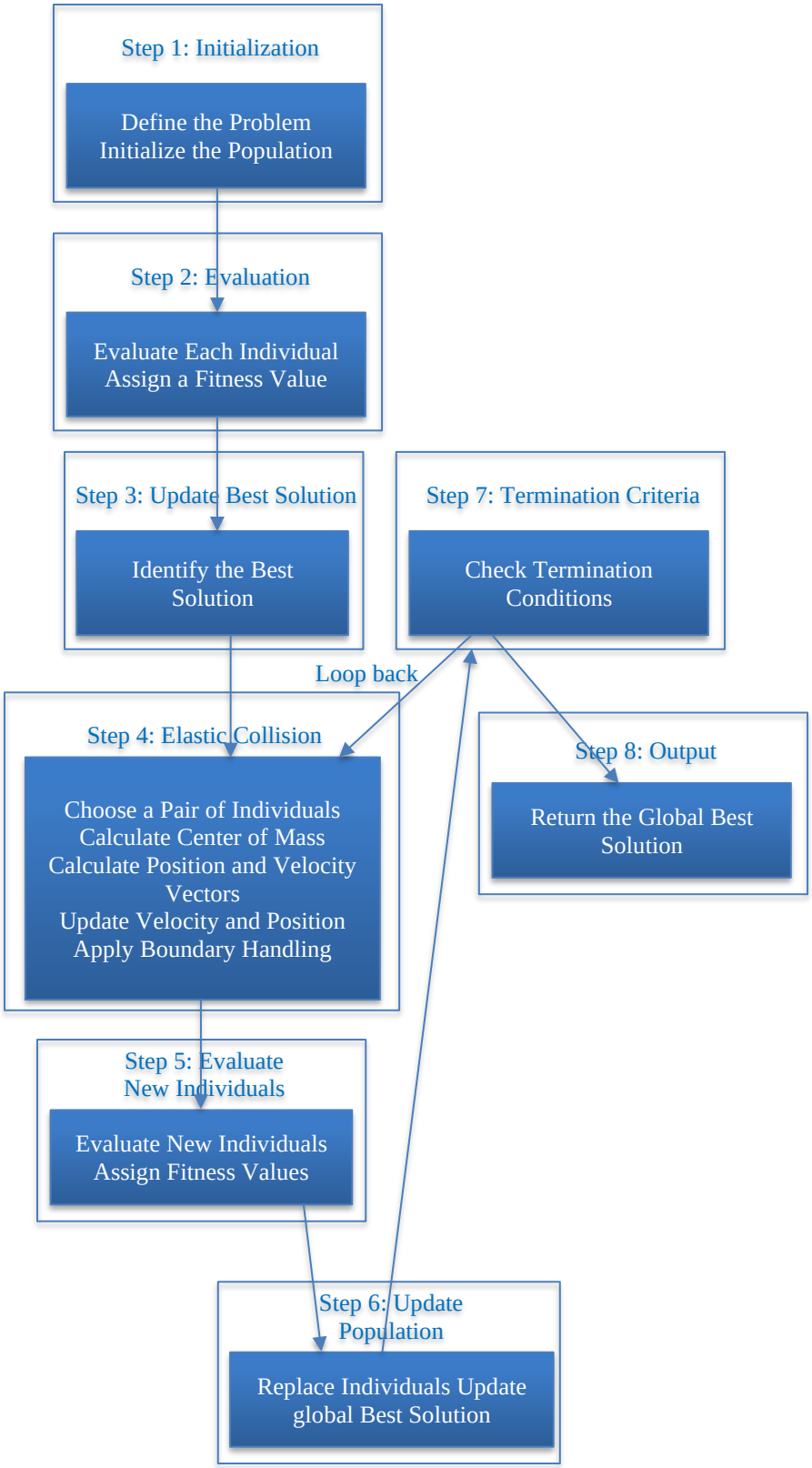

Fig. 4 Effective seeker optimization flowchart

Table 3. Algorithm of ESO optimization

Step 1: Initialization:
- Define the problem to be optimized, including the objective function and the search space.
- Initialize the population of candidate solutions, typically represented as a set of individuals or particles.

Step 2: Evaluation:
- Evaluate each individual in the population by applying the objective function to determine their fitness or quality.
- Assign a fitness value to each individual based on their objective function evaluation.

Step 3: Update Best Solution: Identify the individual with the best fitness value as the global best solution.

Step 4: Elastic Collision:
- Choose a pair of individuals randomly from the population.
- Calculate the center of mass of the selected individuals.
- Calculate the relative position and velocity vectors between the selected individuals.
- Update the velocity vectors of the selected individuals using the collision formula, which involves the mass and velocity of the individuals.
- Update the position vectors of the selected individuals using the updated velocity vectors.
- If necessary, apply appropriate boundary-handling techniques to ensure the position vectors remain within the search space bounds.

Step 5: Evaluate New Individuals:
- Evaluate the fitness of the newly created individuals resulting from the elastic collision operation.
- Assign fitness values to the new individuals.

Step 6: Update Population:
- Replace individuals in the population with the newly created individuals if they have better fitness values.
- Update the global best solution if a new best solution is found.

Step 7: Termination Criteria:
- Check termination conditions, such as the maximum number of iterations or a satisfactory solution quality.
- If the termination conditions are met, stop the algorithm; otherwise, go back to Step 4.

Step 8: Output:
- Return the global best solution found as the optimized solution to the problem.

Every seeker in the same subpopulation has access to the same information. As was discussed before, the inertia weight notation is implemented to shorten the step length as the number of iterations continues, which in turn helps to enhance the search correctness progressively. To improve the efficiency of the local search technique and to include some element of randomness in the search operation, the value u_i for the i^{th} seeker is determined independently for each variable that constitutes a solution.

$$u_{i,j} = rnd(u_i, 1)\ for\ j = 1,2, \dots, D \quad (20)$$

The fuzzy reasoning's action component, which establishes the step length $a_{i,j}$ for the j^{th} variable of the i^{th} seeker is where $rnd(u_i, 1)$ a random number is within the range of $u_{i,j}$.where $u_{i,j}$ is a range, and $rnd(u_i, 1)$ is a random integer, where a random number is $rnd(u_i, 1)$.

$$a_{i,j} = \delta_j * \sqrt{-In(u_{i,j})} \quad (21)$$

The current location of each seeker, as updated i and the mutable j associated with the issue, is calculated using step length a and direction vector d.

$$x_{i,j}(t+1) = x_{i,j}(t) + a_{i,j}(t) * d_{i,j}(t) \quad (22)$$

The ESO algorithm's primary search is found in equation (22). Subpopulations continue to gain knowledge from one another with each passing generation. Inter-subpopulation learning is the term used to describe this phenomenon. In the first implementation, the current positions held by the two persons in each subpopulation regarded as the worst are swapped with those held by the individuals deemed the best in each of the other two subpopulations at each generation. This ensures that the positions held by the people considered the worst in each subpopulation are filled by those considered the best in the other two subpopulations.

The method can potentially display premature convergence and get trapped in a locally optimal solution if the search process of each subpopulation is entirely directed by the information available locally. Inter-subpopulation learning and information sharing make it possible to avoid early convergence and speed up the journey of the ESO to the most accurate feature extraction. This is made possible by the fact that it is possible to prevent premature convergence.

4. Results and Discussion

This section presents the results of the simulation and makes a performance comparison between the suggested approach and other methods that are already in use. This section evaluates the performance for many users and several message sizes spanning a range of durations.

4.1. Performance Evaluation

Table 4 compares the encryption and decryption time of various methods. Here, the proposed OQLC method resulted in reduced encryption and decryption time as compared to ReonV [11], uBlock [12], LWC [13], and HLCAS [15]. Figure 5 shows the graphical representation of Table 4. The Proposed OQLC method shows approximately a 77.09% improvement in encryption time and an 87.68% improvement in decryption time compared to the ReonV method.

The Proposed OQLC method demonstrates around a 78.10% improvement in encryption time and an 85.44% improvement in decryption time compared to the uBlock method. The Proposed OQLC method exhibits approximately a 70.14% improvement in encryption time and a 77.36% improvement in decryption time compared to the LWC method.

The Proposed OQLC method is approximately 66.11% faster in encryption time and 72.62% faster in decryption time than the HLCAS method.

Table 4. Comparison of various techniques' encryption and decryption times

Method	Encryption Time (seconds)	Decryption Time (seconds)
ReonV [11]	0.00441	0.00284
uBlock [12]	0.00461	0.00158
LWC [13]	0.00345	0.00106
HLCAS [15]	0.00301	0.00084
Proposed OQLC	0.00101	0.00023

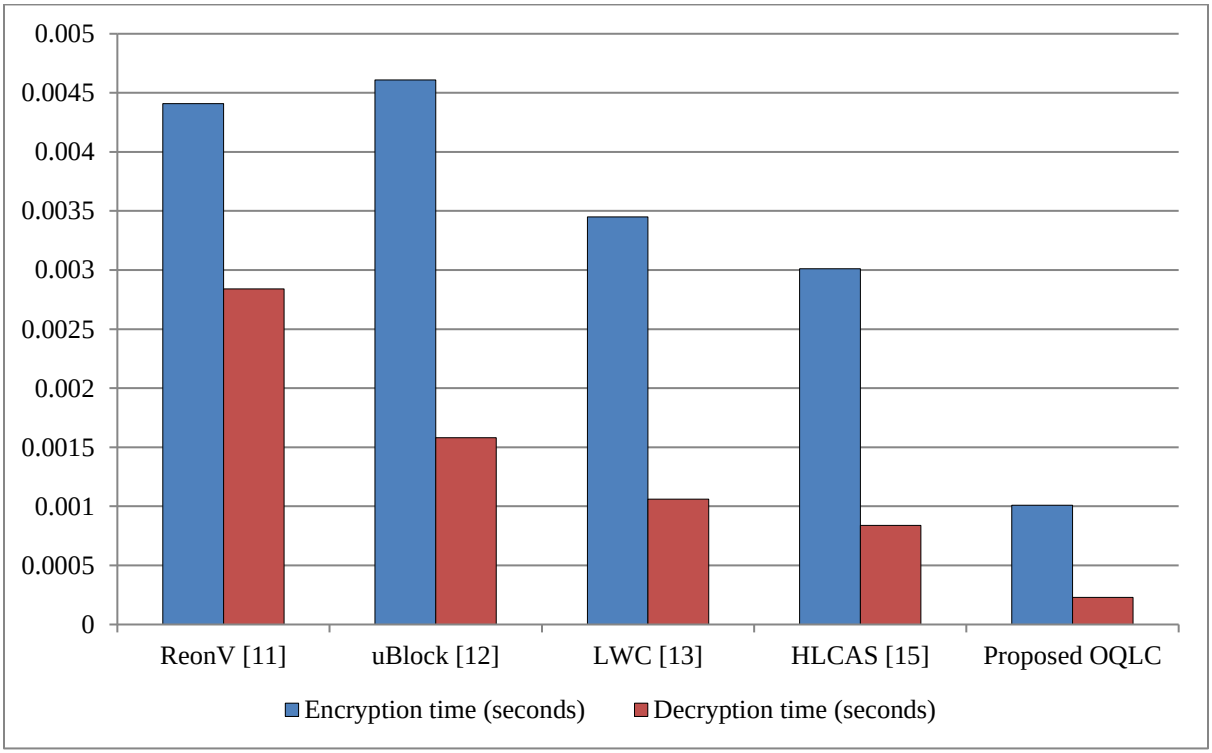

Fig. 5 Comparision of various techniques' encryption and decryption times

4.2. Performance Evaluation for Multiple Users

Table 5 compares the encryption time comparison of multiple user scenarios. Here, the proposed OQLC resulted in reduced encryption time as compared to HLF [18], BLAKE2s [19], LWC-ABE [20], and ELCD [25]. Figure 6 shows the graphical representation of Table 5.

For User 1, the OQLC method has the lowest time and shows a 45.45% improvement compared to the HLF method, a -57.53% decrease compared to the LWC-ABE method, a -4.65% decrease compared to the ELCD method, and a 0.53% development equated to the BLAKE2s method. For User 2, the proposed OQLC method has the lowest time and shows a 39.02% improvement compared to the HLF method, a -42.11% decrease compared to the LWC-ABE method, a -68.42% decrease compared to the ELCD method, and a 271.43% development equated to the BLAKE2s method.

For User 3, the Proposed OQLC method has the lowest time and shows a 10.71% improvement compared to the HLF method, a -32.43% decrease compared to the LWC-ABE method, a -1.75% decrease compared to the ELCD method, and an 8.33% improvement compared to the BLAKE2s method. For User 4, the Proposed OQLC method has the lowest time and shows a 30.56% improvement compared to the HLF method, a -48.81% decrease compared to the LWC-ABE method, a -6.10% decrease compared to the ELCD method, and a 3.03% development equated to the BLAKE2s method. For User 5, The Proposed OQLC method has the lowest time. It shows a 15.38% improvement compared to the HLF method, a -21.79% decrease compared to the LWC-ABE method, a 103.85% development equated to the ELCD method, and a 2.63% development equated to the BLAKE2s method. For User 6, The Proposed OQLC method has the lowest time and shows a 46.15%

improvement compared to the HLF method, a 12.90% improvement compared to the LWC-ABE method, a -3.23% decrease compared to the ELCD method, and a -5.88% decrease compared to the BLAKE2s method. For User 7, The Proposed OQLC method has the lowest time and shows a -63.16% decrease compared to the HLF method, a 5.71% improvement compared to the LWC-ABE method, a -26.67% decrease compared to the ELCD method, and a 13.16% development equated to the BLAKE2s method.

For User 8, The Proposed OQLC method has the lowest time and shows a -35.09% decrease compared to the HLF method, a -45.96% decrease compared to the LWC-ABE method, a -0.48% decrease compared to the ELCD method, and a 91.23% development equated to the BLAKE2s method.

For User 9, The Proposed OQLC method has the lowest time. It shows a -39.13% decrease compared to the HLF method, a -15.38% decrease compared to the LWC-ABE method, a 7.69% development equated to the ELCD method, and a 36.11% development equated to the BLAKE2s method. For User 10, The Proposed OQLC method has the lowest time and shows a 61.76% improvement compared to the HLF method.

Table 5. Encryption time comparison of multiple user scenarios

Users	HLF [18]	BLAKE2s [19]	LWC-ABE [20]	ELCD [25]	Proposed OQLC
1	0.0033	0.0027	0.0043	0.0022	0.0019
2	0.0025	0.0041	0.0019	0.0038	0.0011
3	0.0028	0.0025	0.0037	0.0018	0.0012
4	0.0054	0.0039	0.0042	0.0041	0.0022
5	0.0032	0.0027	0.0039	0.0013	0.0014
6	0.0018	0.0034	0.0031	0.0032	0.0017
7	0.0051	0.0042	0.0035	0.0026	0.0019
8	0.0042	0.0057	0.0047	0.0037	0.0023
9	0.0046	0.0036	0.0026	0.0041	0.0028
10	0.0055	0.0034	0.0034	0.0043	0.0015

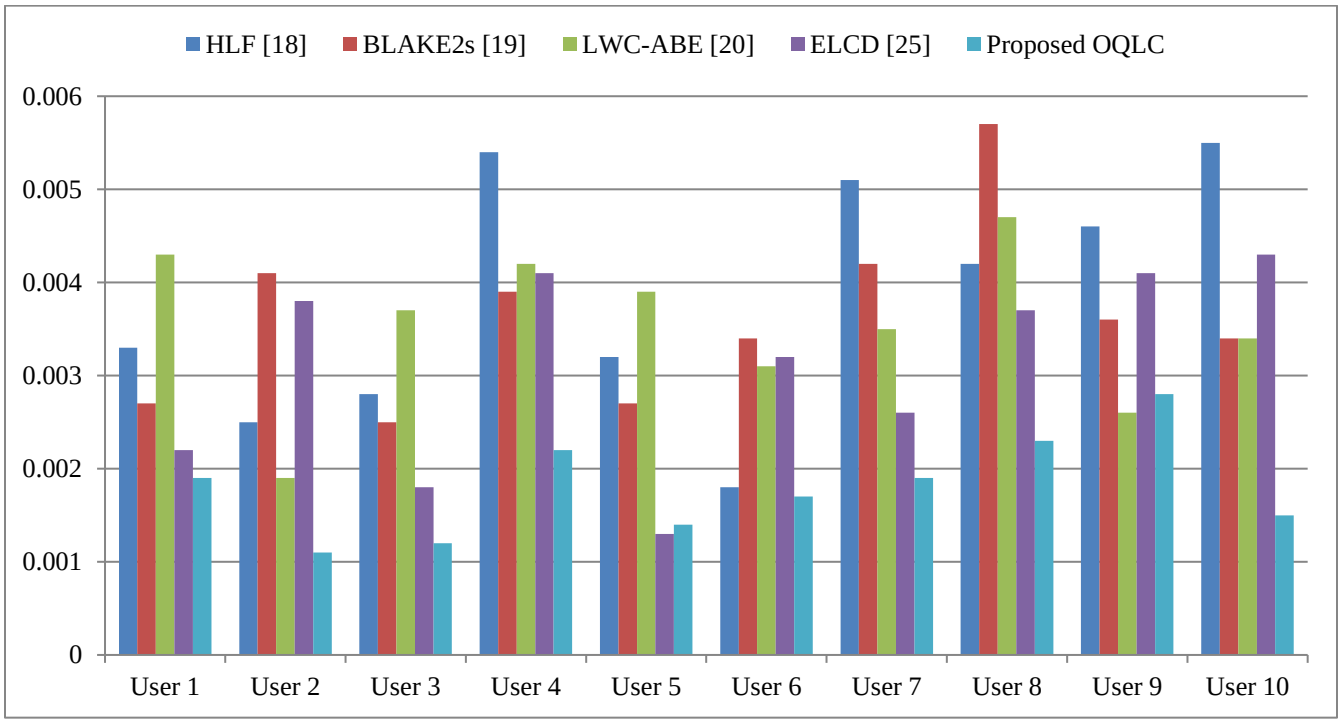

Fig. 6 Encryption time comparison of multiple user scenarios

Table 6. Decryption time comparison of multiple user scenarios

Users	HLF [18]	BLAKE2s [19]	LWC-ABE [20]	ELCD [25]	Proposed OQLC
1	0.0048	0.0044	0.0041	0.0028	0.0023
2	0.0049	0.0048	0.0043	0.0026	0.0014
3	0.0049	0.0048	0.0046	0.0033	0.0019
4	0.0048	0.0046	0.0044	0.0032	0.0013
5	0.0049	0.0047	0.0044	0.0030	0.0014
6	0.0049	0.0048	0.0047	0.0034	0.0027
7	0.0047	0.0046	0.0042	0.0036	0.0016
8	0.0048	0.0046	0.0045	0.0032	0.0028
9	0.0049	0.0047	0.0044	0.0033	0.0015
10	0.0048	0.0046	0.0045	0.0031	0.0016

Table 6 compares the decryption time comparison of multiple user scenarios. Here, the proposed OQLC resulted in reduced decryption time as compared to HLF [18], BLAKE2s [19], LWC-ABE [20], and ELCD [25]. Figure 7 shows the graphical representation of Table 5. For User 1, The ELCD method has the lowest time and shows a -52.17% decrease compared to the HLF method, a -47.83% decrease compared to the BLAKE2s method, a -36.59% decrease compared to the LWC-ABE method, and a -34.78% decrease compared to the proposed OQLC method. For User 2, the Proposed OQLC method has the lowest time and shows a 71.43% improvement compared to the HLF method, a 7.14% improvement compared to the BLAKE2s method, a -67.44% decrease compared to the LWC-ABE method, and a 35.71% improvement compared to the ELCD method. For User 3, the Proposed OQLC method has the lowest time. It shows a 61.22% improvement compared to the HLF method, a 53.06% development equated to the BLAKE2s method, a 26.09% development equated to the LWC-ABE method, and a 45.45% improvement compared to the ELCD method.

For User 4, the Proposed OQLC method has the lowest time. It shows a 47.92% development equated to the HLF method, a 41.67% development equated to the BLAKE2s method, a 22.73% development equated to the LWC-ABE method, and a 57.69% improvement compared to the ELCD method. For User 5, the Proposed OQLC method has the lowest time. It shows a 48.98% development equated to the HLF method, a 40.43% improvement compared to the BLAKE2s method, a 29.55% improvement compared to the LWC-ABE method, and a 53.33% development equated to the ELCD method.

For User 6, the Proposed OQLC method has the lowest time. It shows a 43.75% improvement compared to the HLF method, a 33.33% improvement compared to the BLAKE2s method, a 21.28% development equated to the LWC-ABE method, and a 37.04% improvement compared to the ELCD method. For User 7, The Proposed OQLC method has the lowest time. It shows a 65.96% improvement compared to the HLF method, a 58.70% improvement compared to the BLAKE2s method, a 19.05% development equated to the LWC-ABE method, and a 55.56% development equated to the ELCD method.

For User 8, The Proposed OQLC method has the lowest time. It shows a 40.63% development equated to the HLF method, a 43.75% improvement compared to the BLAKE2s method, a 37.78% improvement compared to the LWC-ABE method, and a 54.55% development equated to the ELCD method. For User 9, The Proposed OQLC method has the lowest time. It shows a 31.91% improvement compared to the HLF method, a 34.04% improvement compared to the BLAKE2s method, a 31.82% development equated to the LWC-ABE method, and a 54.55% development equated to the ELCD method. For User 10, The Proposed OQLC method has the lowest time and shows a 37.50% improvement compared to the HLF method,

4.3. Performance Evaluation for Multiple Message Lengths

Table 7 compares the encryption time comparison of multiple message length scenarios. Here, the proposed OQLC resulted in reduced encryption time as compared to TEA [26], DESL [27], AES-RSA [28], and AES-ECC [29]. Figure 8 shows the graphical representation of Table 7. For a message length of 1000, The Proposed OQLC method has the lowest time. It shows a 67.60% improvement compared to the TEA method, a 66.07% development equated to the DESL method, a 65.74% development equated to the AES-RSA method, and a 65.01% improvement compared to the AES-ECC method. For a message length of 900, The Proposed OQLC method has the lowest time. It shows a 68.86% development equated to the TEA method, a 68.18% improvement compared to the DESL method, a 67.70% development equated to the AES-RSA method, and a 66.94% development equated to the AES-ECC method. For a message length of 400, The Proposed OQLC method has the lowest time. It shows a 69.03% development equated to the TEA method, a 68.49% development equated to the DESL method, a 66.77% development equated to the AES-RSA method, and a 63.29% development equated to the AES-ECC method.

Table 7. Encryption time comparison of multiple message length scenarios

ML	TEA [26]	DESL	AES-RSA [28]	AES-ECC [29]	Proposed OQLC
1000	0.00343	0.00336	0.00324	0.00317	0.00111
900	0.00334	0.00330	0.00319	0.00310	0.00105
800	0.00329	0.00326	0.00316	0.00308	0.00100
700	0.00325	0.00324	0.00314	0.00304	0.00199
600	0.00322	0.00321	0.00311	0.00302	0.00193
500	0.00320	0.00319	0.00308	0.00301	0.00192
400	0.00317	0.00318	0.00306	0.00297	0.00189
300	0.00315	0.00314	0.00303	0.00294	0.00188
200	0.00312	0.00312	0.00301	0.00293	0.00183
100	0.00311	0.00308	0.00299	0.00291	0.00180

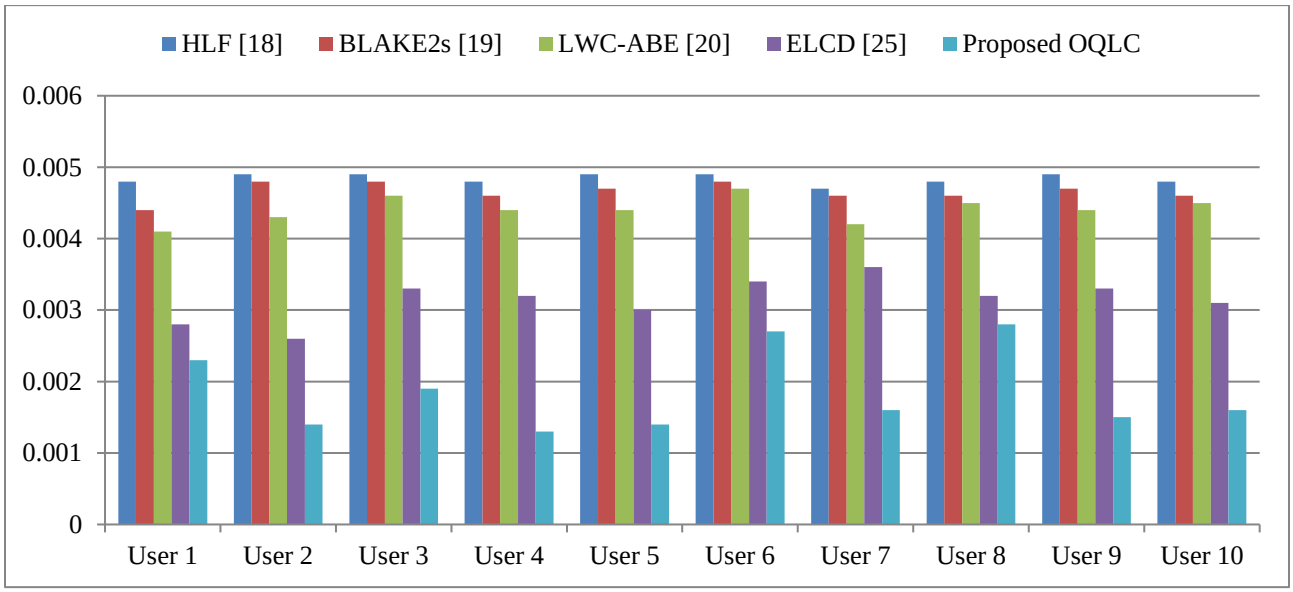

Fig. 7 Decryption time comparison of multiple user scenarios

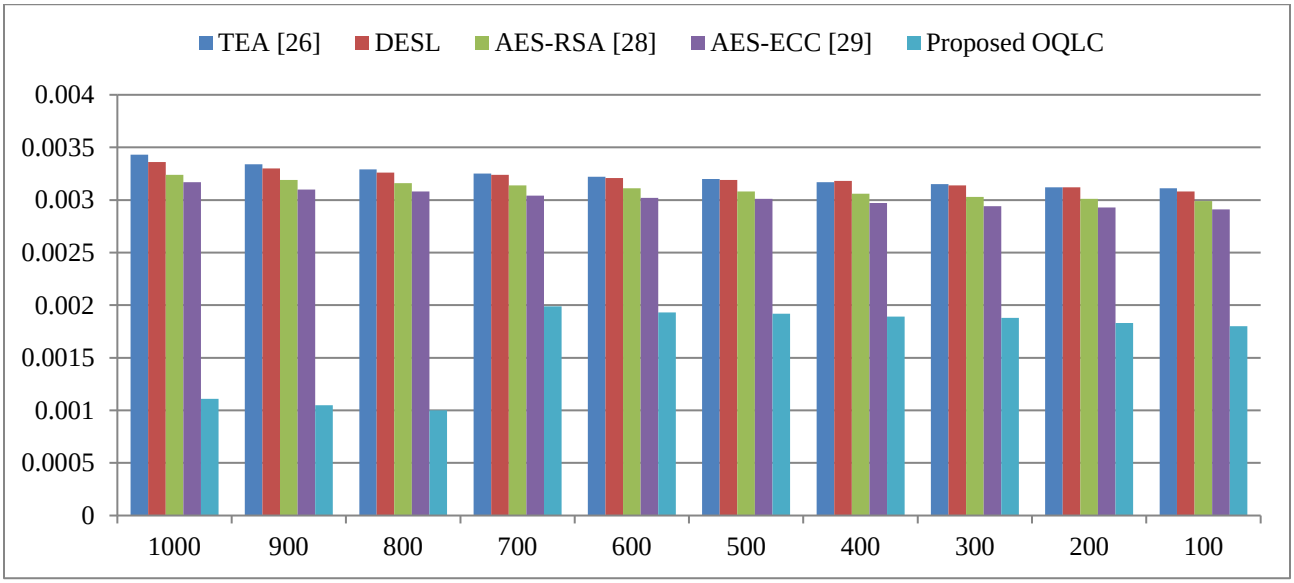

Fig. 8 Encryption time comparison of multiple message length scenarios

For a message length of 300, The Proposed OQLC method has the lowest time. It shows a 68.85% development equated to the TEA method, a 69.08% development equated to the DESL method, a 66.40% development equated to the AES-RSA method, and a 61.46% development equated to the AES-ECC method. For a message length of 200, The Proposed OQLC method has the lowest time. It shows a 68.59% development equated to the TEA method, a 68.59% development equated to the DESL method, a 65.28% improvement compared to the AES-RSA method, and a 62.68% development equated to the AES-ECC method. For a message length of 100, The Proposed OQLC method has the lowest time. It shows a 66.67% development equated to the TEA method, a 69.10% development equated to the DESL method, a 65.55% development equated to the AES-RSA method, and a 61.82% development equated to the AES-ECC method. Table 8 compares the decryption time comparison of multiple message length scenarios. Here, the proposed OQLC resulted in reduced decryption time as compared to TEA [26], DESL [27], AES-RSA [28], and AES-ECC [29]. Figure 9 shows the graphical representation of

For a message length of 1000, the Proposed OQLC method has the lowest time. It shows a 46.11% development equated to the TEA method, a 46.11% development equated to the DESL method, a 45.99% development equated to the AES-RSA method, and a 45.68% development equated to the AES-ECC method. For a message length of 900, The Proposed OQLC method has the lowest time. It shows a 47.70% development equated to the TEA method, a 47.70% development equated to the DESL method, a 47.74% development equated to the AES-RSA method, and a 46.47% improvement compared to the AES-ECC method. For a message length of 800, The Proposed OQLC method has the lowest time. It shows a 48.84% development equated to the TEA method, a 48.84% development equated to the DESL method, a 48.07% development equated to the AES-RSA method, and a 48.07% development equated to the AES-ECC method. For a message length of 700, The Proposed OQLC method has the lowest time. It shows a 50.12% development equated to the TEA method, a 50.12% development equated to the DESL method, a 49.75% development equated to the AES-RSA method, and a 49.75% development equated to the AES-ECC method.

For a message length of 600, The Proposed OQLC method has the lowest time. It shows a 25.46% development equated to the TEA method, a 25.46% development equated to the DESL method, a 25.68% development equated to the AES-RSA method, and a 25.46% development equated to the AES-ECC method. For a message length of 500, The Proposed OQLC method has the lowest time. It shows a 31.23% development equated to the TEA method, a 31.23% development equated to the DESL method, a 31.16% development equated to the AES-RSA method, and a 30.84% improvement compared to the AES-ECC method. For a message length of 400, The Proposed OQLC method has the lowest time. It shows a 31.47% development equated to the TEA method, a 31.47% development equated to the DESL method, a 31.19% development equated to the AES-RSA method, and a 30.64% development equated to the AES-ECC method.

For a message length of 300, The Proposed OQLC method has the lowest time. It shows a 29.03% development equated to the TEA method, a 29.47% development equated to the DESL method, a 28.63% development equated to the AES-RSA method, and a 28.25% development equated to the AES-ECC method. For a message length of 200, The Proposed OQLC method has the lowest time. It shows a 28.89% development equated to the TEA method, a 29.17% development equated to the DESL method, a 29.40% development equated to the AES-RSA method, and a 29.40% development equated to the AES-ECC method. For a message length of 100, The Proposed OQLC method has the lowest time. It shows a 48.71% development equated to the TEA method, a 48.43% improvement compared to the DESL method, a 46.92% development equated to the AES-RSA method, and a 45.99% development equated to the AES-ECC method.

Table 8. Decryption time comparison of multiple message length scenarios

ML	TEA [26]	DESL	AES-RSA [28]	AES-ECC [29]	Proposed OQLC
1000	0.00443	0.00443	0.00442	0.00440	0.00239
900	0.00428	0.00427	0.00427	0.00425	0.00224
800	0.00415	0.00414	0.00412	0.00412	0.00212
700	0.00407	0.00405	0.00404	0.00404	0.00204
600	0.00392	0.00392	0.00391	0.00391	0.00291
500	0.00386	0.00385	0.00385	0.00384	0.00282
400	0.00375	0.00375	0.00374	0.00373	0.00273
300	0.00364	0.00362	0.00360	0.00359	0.00259
200	0.00356	0.00354	0.00352	0.00352	0.00251
100	0.00351	0.00351	0.00350	0.00349	0.00249

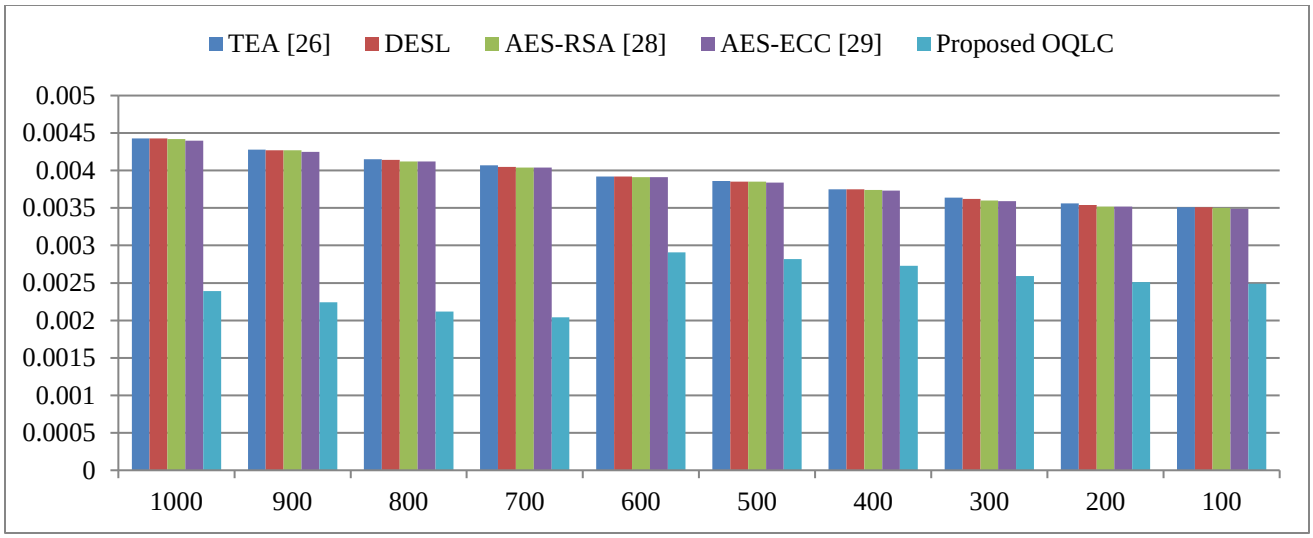

Fig. 9 Decryption time comparison of multiple message length scenarios

5. Conclusion

This research meticulously examined a novel and innovative approach known as OQLC. The primary goal of OQLC was to establish a framework that could ensure the secure transmission and safeguarding of sensitive data, even in the most resource-constrained environments. The OQLC algorithm was a brilliant fusion of two distinct but complementary elements: the ERG-128 algorithm, which belongs to the realm of ULWC, and the RQKD-QC framework, which is rooted in the domain of quantum cryptography. The rationale behind this hybridization was to leverage the lightweight efficiency of ERG-128 and the security prowess of quantum cryptography to create a robust cryptographic solution that could cater to the multifaceted needs of the modern IoT ecosystem. To further elevate the performance and efficiency of the OQLC algorithm, an ingenious technique known as ESO was harnessed. ESO capitalized on natural seekers' inherent parallelism and adaptability to fine-tune crucial aspects of the cryptographic process. It delved into optimizing key scheduling, round functions, and cryptographic primitives within the OQLC framework. The result was an exquisite equilibrium between lightweight implementation, post-quantum security, and remarkable cryptographic performance. One of the standout achievements of the OQLC algorithm was its remarkable efficiency in terms of message processing. In a practical context, when dealing with a message length of 100, the OQLC method outperformed existing techniques by a substantial margin. It exhibited a 48.71% enhancement compared to the TEA method, a 48.43% improvement over the DESL method, a 46.92% development relative to the AES-RSA method, and a 45.99% improvement when compared to the AES-ECC method. Further research could explore and develop even more advanced optimization methods in this domain. Additionally, incorporating techniques such as watermarking and steganography schemes could bolster the security and functionality of lightweight cryptographic systems, making them even more versatile and resilient in the face of evolving IoT challenges. This ongoing exploration and innovation are crucial to maintaining the integrity and security of data transmitted and processed within the IoT ecosystem.

References

[1] Rana Abbas Al-Kaabi, and Alharith A. Abdullah, "A Survey: Medical Health Record Data Security Based on Interplanetary File System and Blockchain Technologies," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 30, no. 1, pp. 586-597, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] Jan Carlo T. Arroyo et al., "An Enhanced Playfair Algorithm with Dynamic Matrix Using the Novel Multidimensional Element-in-Grid Sequencer (MEGS)," *International Journal of Engineering Trends and Technology*, vol. 70, no. 3, pp. 132-139, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[3] Neha Sharma, Chinmay Chakraborty, and Rajeev Kumar, "Optimized Multimedia Data through Computationally Intelligent Algorithms," *Multimedia Systems*, vol. 29, no. 5, pp. 2961-2977, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[4] Nasir N. Hurrah, Ekram Khan, and Uzma Khan, "CADEN: Cellular Automata and DNA Based Secure Framework for Privacy Preserving in IoT Based Healthcare," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 3, pp. 2631-2643, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] Joseph B. Awotunde et al., "An IoMT-Based Steganography Model for Securing Medical Information," *International Journal of Healthcare Technology and Management*, vol. 19, no. 3-4, pp. 218-236, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Swetha Pesaru, Naresh K. Mallenahalli, and B. Vishnu Vardhan, "Light Weight Cryptography-Based Data Hiding System for Internet of Medical Things," *International Journal of Healthcare Management*, pp. 1-14, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[7] Zied Guitouni, Mohammed Ali Ghaieb, and Mohsen Machhout, "Security Analysis of Medical Image Encryption Using AES Modes for IoMT Systems," *International Journal of Computer Applications*, vol. 185, no. 2, pp. 15-21, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[8] Wadood Abdul, "Security of Medical Images Over Insecure Communication Channels Using Zero-Steganography," *International Journal of Distributed Sensor Networks*, vol. 18, no. 2, pp. 1-10, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] Suyel Namasudra, "A Secure Cryptosystem Using DNA Cryptography and DNA Steganography for the Cloud Based IoT Infrastructure," *Computers and Electrical Engineering*, vol. 104, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] Oluwakemi Christiana Abikoye et al., "Securing Critical User Information over the Internet of Medical Things Platforms Using a Hybrid Cryptography Scheme," *Future Internet*, vol. 15, no. 3, pp. 1-33, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Wajih El Hadj Youssef et al., "An Efficient Lightweight Cryptographic Instructions Set Extension for IoT Device Security," *Security and Communication Networks*, vol. 2022, no. 1, pp. 1-17, 2022. [CrossRef] [Google Scholar] [Publisher Link]

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

[13] V. Panchami, and Mahima Mary Mathews, "A Substitution Box for Lightweight Ciphers to Secure Internet of Things," *Journal of King Saud University-Computer and Information Sciences*, vol. 35, no. 4, pp. 75-89, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Mohammed El-hajj, Hussien Mousawi, and Ahmad Fadlallah, "Analysis of Lightweight Cryptographic Algorithms on IoT Hardware Platform," *Future Internet*, vol. 15, no. 2, pp. 1-29, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] P. Prakasam et al., "Low Latency, Area and Optimal Power Hybrid Lightweight Cryptography Authentication Scheme for Internet of Things Applications," *Wireless Personal Communications*, vol. 126, no. 1, pp. 351-365, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[16] Susila Windarta et al., "Lightweight Cryptographic Hash Functions: Design Trends, Comparative Study, and Future Directions," *IEEE Access*, vol. 10, pp. 82272-82294, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[17] Agus Winarno, and Riri Fitri Sari, "A Novel Secure End-to-End IoT Communication Scheme Using Lightweight Cryptography Based on Block Cipher," *Applied Sciences*, vol. 12, no. 17, pp. 1-19, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Suhair Alshehri, and Omaimah Bamasag, "AAC-IoT: Attribute Access Control Scheme for IoT Using Lightweight Cryptography and Hyperledger Fabric Blockchain," *Applied Sciences*, vol. 12, no. 16, pp. 1-20, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[19] Rizki Agus Zandra Kurniawan, Sri Wahjuni, and Shelvie Nidya Neyman, "Secure Communication Protocol for Arduino-Based IoT Using Lightweight Cryptography," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 12, no. 2, pp. 453-459, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[20] Mounika Jammula, Venkata Mani Vakamulla, and Sai Krishna Kondoju, "Hybrid Lightweight Cryptography with Attribute-Based Encryption Standard for Secure and Scalable IoT System," *Connection Science*, vol. 34, no. 1, pp. 2431-2447, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[21] Soline Blanc et al., "Benchmarking of Lightweight Cryptographic Algorithms for Wireless IoT Networks," *Wireless Networks*, vol. 28, no. 8, pp. 3453-3476, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[22] Kyriaki Tsantikidou, and Nicolas Sklavos, "Hardware Limitations of Lightweight Cryptographic Designs for IoT in Healthcare," *Cryptography*, vol. 6, no. 3, pp. 1-13, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[23] Israa Ezzat Salem, Haider Rasheed Abdulshaheed, and Hassan Muwafaq Gheni, "A Secure Telemedicine Electronic Platform Based on Lightweight Cryptographic Approach," *Telecommunication Computing Electronics and Control*, vol. 20, no. 5, pp. 988-995, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[24] Tarun Kumar Goyal, Vineet Sahula, and Deepak Kumawat, "Energy Efficient Lightweight Cryptography Algorithms for IoT Devices," *IETE Journal of Research*, vol. 68, no. 3, pp. 1722-1735, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[25] Adel A. Ahmed, and Omar M. Barukab, "Unforgeable Digital Signature Integrated into Lightweight Encryption Based on Effective ECDH for Cybersecurity Mechanism in Internet of Things," *Processes*, vol. 10, no. 12, pp. 1-27, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[26] Hakeem Imad Mhaibes, May Hattim Abood, and Alaa Kadhim Farhan, "Simple Lightweight Cryptographic Algorithm to Secure Imbedded IoT Devices," *International Journal of Interactive Mobile Technologies*, vol. 16, no. 20, pp. 98-113, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[27] Nari Im, Soyeon Choi, and Hoyoung Yoo, "S-Box Attack Using FPGA Reverse Engineering for Lightweight Cryptography," *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 25165-25180, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[28] Hussein M. Mohammad, and Alharith A. Abdullah, "Enhancement Process of AES: A Lightweight Cryptography Algorithm-AES for Constrained Devices," *Telecommunication Computing Electronics and Control*, vol. 20, no. 3, pp. 551-560, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[29] Ana Goulart et al., "On Wide-Area IoT Networks, Lightweight Security and Their Applications-A Practical Review," *Electronics*, vol. 11, no. 11, pp. 1-40, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[30] Shubham Gupta, and Sandeep Saxena, *Lightweight Cryptographic Techniques and Protocols for IoT*, Internet of Things, pp. 55-77, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[31] Hemant Mahajan, and K.T.V. Reddy, "Secure Gene Profile Data Processing Using Lightweight Cryptography and Blockchain," *Cluster Computing*, vol. 27, no. 3, pp. 2785-2803, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[32] Mohammed El-Hajj, Hussien Mousawi, and Ahmad Fadlallah, "Analysis of Lightweight Cryptographic Algorithms on IoT Hardware Platform," *Future Internet*, vol. 15, no. 2, pp. 1-29, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[33] Vishal A. Thakor et al., "A Novel 5-Bit S-Box Design for Lightweight Cryptography Algorithms," *Journal of Information Security and Applications*, vol. 73, pp. 1-11, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[34] Jonas Gava et al., "Assessment of Radiation-Induced Soft Errors on Lightweight Cryptography Algorithms Running on a Resource-Constrained Device," *IEEE Transactions on Nuclear Science*, vol. 70, no. 8, pp. 1805-1813, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[35] Issam W. Damaj, Hadi Al-Mubasher, and Mahmoud Saadeh, "An Extended Analytical Framework for Heterogeneous Implementations of Light Cryptographic Algorithms," *Future Generation Computer Systems*, vol. 141, pp. 154-172, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[36] Jizheng Xue et al., "Side-Channel Attack of Lightweight Cryptography Based on MixColumn: Case Study of PRINCE," *Electronics*, vol. 12, no. 3, pp. 1-21, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[37] Navdeep Lata, and Raman Kumar, "DSIT: A Dynamic Lightweight Cryptography Algorithm for Securing Image in IOT Communication," *International Journal of Image and Graphics*, vol. 23, no. 4, pp. 1-22, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[38] Nahla Ibrahim, and Johnson Agbinya, "Design of a Lightweight Cryptographic Scheme for Resource-Constrained Internet of Things Devices," *Applied Sciences*, vol. 13, no. 7, pp. 1-17, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[39] Gamil R.S. Qaid, and Nadhem Sultan Ebrahim, "A Lightweight Cryptographic Algorithm Based on DNA Computing for IoT Devices," *Security and Communication Networks*, vol. 2023, no. 1, pp. 1-12, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[40] Muntaser Al-Moselly, and Ali Al-Haj, "High-Performance Hardware Implementation of the KATAN Lightweight Cryptographic Cipher," *Journal of Circuits, Systems and Computers*, vol. 32, no. 1, 2023. [CrossRef] [Google Scholar] [Publisher Link]