International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

UW-LightKey: A Novel Lightweight Key Exchange using Acoustic Channel Characteristics for UWSNs

Nour Zahra, Souheil Khawatmi, Bader Aldin Kassab
Department of Systems and Computer Networks
Faculty of Informatics Engineering
University of Aleppo, Syria

***Abstract* - In this work, we present an innovative scheme called UW-LightKey for the purpose of key exchange within Underwater Wireless Sensor Networks (UWSNs). Contrary to existing public-key cryptography schemes based on computationally intensive algorithms like RSA, ECC, and Diffie-Hellman, UW-LightKey takes advantage of two physical attributes specific to the underwater scenario, namely, acoustic propagation delay and the three-dimensional location of sensor nodes. Both these dynamic properties help in providing entropy for nonce creation, thus avoiding the use of cryptographic heavyweights. UW-LightKey uses only lightweight operations such as XOR, rotate, and the Enhanced Lightweight Hash (ELH). It does not require sending any information about the generated shared secret over the medium and thus avoids any potential threat of eavesdropping. Results of a security evaluation show UW-LightKey's resistance against underwater attack scenarios, while the performance study reveals a key generation time of 243 microseconds compared to 927 (ECC), 2,153 (DH), and 3,634 microseconds (RSA).**

Keywords - Underwater Wireless Sensor Networks (UWSNs); Key Exchange Algorithm; Lightweight Cryptography; Security; Acoustic Channel; Resource-Constrained Networks.

I. INTRODUCTION

There is rapid growth in Underwater Wireless Sensor Networks (UWSNs) because of rising interest in their applications like environmental protection (climate change monitoring), maritime exploitation, surveillance, and pollution monitoring [1]. Nonetheless, the unique properties of the underwater environment impose a series of challenges, among which we could highlight high propagation delay due to the use of acoustic wave propagation compared to radio waves [2], low bandwidth affecting the transmission speed of information, scarce energy resources due to difficulties in charging batteries, and low computational power due to small-scale processors with limited memory space [3].

One of the major challenges within this field is the issue of network security since the open and hostile underwater environment exposes the network to various types of security attacks, like eavesdropping, man-in-the-middle, replay, and energy drain attacks [2]. The key exchange process constitutes the basis for ensuring secure communications; Nonetheless, traditional key exchange algorithms such as Diffie-Hellman (DH) and RSA are computationally intensive and energy-consuming, thus impractical in resource-constrained nodes of underwater communication networks [4]. As shown by recent research work about security in UWSNs [5, 6], these issues have been systematically identified and increasingly addressed through novel lightweight cryptographic approaches that provide adequate security in an energy-constrained setting.

Hence, there is an increasing demand for efficient algorithms that ensure security but are energy and computationally efficient at the same time. In this regard, this paper presents a lightweight key exchange algorithm, called UW-LightKey, which utilizes innovative principles like the application of randomness derived from dynamic properties of the underwater acoustic channel and the three-dimensional positioning of sensors as a means of generating a secret key. For generating the hash value, the Enhanced Lightweight Hash (ELH) algorithm is used, which draws upon SPARKLE and PHOTON functions designed for software execution on limited hardware devices.

The goals of the algorithm include:

- Minimizing computation through simple operations like XOR and bitwise rotation.
- Minimizing energy consumption through compact data exchanges and minimal cryptographic operations.
- Ensuring security by associating the key with unique physical parameters that cannot be predicted or replicated.
- Resisting common attacks in the UWSN environment without the need for any sophisticated infrastructure.

The organization of this paper includes Section 2 for literature review, Section 3 for a detailed description of the UW-LightKey algorithm, Section 4 for security analysis, Section 5 for performance evaluation and experiment results, followed by conclusions in Section 6.

II. LITERATURE REVIEW

Securing communications and performing node authentication in Underwater Wireless Sensor Networks (UWSNs) can be considered quite challenging tasks because of strict restrictions concerning computation, communication, and energy, as well as adverse underwater conditions prone to various kinds of attacks. In order to solve this problem, several protocols and schemes for key management were developed, demonstrating a transition from conventional to lightweight and more specific approaches. This part provides a review of related literature highlighting the techniques used and their drawbacks.

Early research was oriented toward using traditional methods based on public-key encryption and signature

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

algorithms such as RSA and Diffie-Hellman (DH). Being rather secure, they cannot be applied directly in UWSNs since they require huge amounts of both energy and computation, which cannot be provided by typical sensor nodes. Though Elliptic Curve Cryptography (ECC) offers a similar security level but with much shorter keys considered more efficient, this technique still includes relatively complex calculations and cannot reduce the communication overhead inherent to high-latency channels.

A. Hash Functions and Symmetric Encryption-Based Protocols

A number of works propose lightweight protocols using hash functions and XOR for achieving mutual authentication and key exchange. For example, [7] offers a hash function protocol to establish mutual authentication and exchange the key in UWSNs. This scheme uses a pre-existing channel for transmitting IDs and passwords, timestamps, and random numbers as protection against replay attacks. Nevertheless, this scheme is resistant neither to node capture attacks nor to forward secrecy and quantum attack resistance.

B. Elliptic Curve Cryptography-Based Protocols

As being more energy efficient than other approaches, ECC is used in a variety of UWSN key management frameworks. One such solution is offered in [8], which describes a lightweight key management system based on ECC and including key generation, distribution, and revocation procedures. This framework ensures high security of the scheme and provides very short keys, though it is vulnerable to node capture attacks.

C. Protocols using Physically Unclonable Functions (PUFs) and Chaotic Maps

Some of the protocols proposed have considered the combination of Physically Unclonable Functions (PUFs) and chaotic maps to counter potential attacks resulting from physical capture. Physically Unclonable Functions (PUFs) are hardware security techniques that rely on unique physical variations in order to create device fingerprints, while chaotic maps refer to mathematical algorithms with sensitive initial conditions and are used extensively in cryptography for generating keys. Reference [9], for instance, introduces a lightweight authentication protocol for Underwater Acoustic Networks (UANs), which uses Physically Unclonable Functions (PUFs) to guarantee protection of secret data stored in gateways and sensor nodes, with chaotic maps being utilized to realize an efficient process of authentication with minimal computational complexity. In doing so, the proposed protocol increases security levels against potential capture attacks; however, it does not provide support for location-based authentication (checking a node's geographical location to verify its identity) and lacks resistance to quantum attacks.

D. Post-Quantum Protocols using Lattice-Based Cryptography and Quantum Resistance

The rising danger posed by quantum computers necessitates the implementation of quantum-resistant protocols. One such protocol has been introduced in reference [10], where a two-round protocol based on the NTRU lattice-based algorithm – a cryptographic method based on difficult problems from lattice theory – has been proposed to provide support for anonymity (the ability to hide node identities from an adversary) and location-based authentication (the ability to verify node identities based on their geographic location). By relying on the lightweight and quantum-resistant features of NTRU lattice-based cryptography, the presented protocol also accounts for node movement in underwater currents; however, it results in increased communication costs when compared to conventional approaches.

E. Channel-Based Key Generation in UWSNs

Another stream of research explores exploiting the inherent randomness of the channel itself to extract secret information used in cryptographic key generation. The multi-hop protocol described in [11] exploits random route propagation delays through the network topology as a source of randomness, effectively transforming the slow speed of sound waves into a security advantage. A conference paper by the same authors describes a key agreement algorithm based on the statistical distribution of channel impulse response (CIR) properties, accounting for the time-varying properties of the underwater channel [12].

The same research group further developed this technique with an all-encompassing CIR-based secret key generation scheme [13], which makes use of extended propagation delays to allow probe transmissions simultaneously and uses K-means quantization to generate 3 times more secret bits compared to standard uniform quantization.

Other physical parameters were considered as alternatives in some studies. For instance, the work in [14] presents a localization scheme using adversarial neural cryptography based on acoustic channel parameters, where randomly selected physical layer parameters are used as dynamic keys while maintaining the localization accuracy. In another example, a lightweight mutual authentication and key exchange protocol for IoUT (Internet of Underwater Things) was proposed [15]; however, this protocol does not consider channel parameters in the protocol operation.

Although promising, channel-based techniques suffer from several drawbacks. First, CIR extraction requires highly dynamic channel conditions to ensure sufficient randomness entropy in the channel parameters. Slow-fading or stationary channels pose a challenge to CIR extraction methods due to a lack of variance. In addition, probing and synchronization with respect to physical channel parameters are required, resulting in increased overhead. Unlike channel-based schemes, UW-LightKey does not depend on channel parameter variation for key randomness. Instead, the physical parameters used in the UW-LightKey protocol include the constant propagation delays and position of three-dimensional nodes and can therefore benefit from more robust conditions. At the same time, UW-LightKey demonstrates higher efficiency with a faster key generation process (243 µs).

The above survey demonstrates that there exists an increasing trend towards specialization in security mechanisms for underwater wireless sensor networks (UWSNs) due to the unique challenges posed by the underwater environment. However, a major problem still lies in the development of a framework capable of incorporating lightweight nature, physical attack resistance, quantum-resistance capabilities, and maintaining efficient communication. This paper seeks to provide a solution for this challenge.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

In order to enable the effective comparison between existing UWSN security mechanisms and the proposed scheme, UW-Lightkey, we have presented two comparative tables. Tab.1 compares cryptographic-based mechanisms based on their computational efficiency and security regarding key delivery. Tab. 2 presents the comparison of channel-based mechanisms based on entropy sources and physical layer awareness.

Table 1 demonstrates a comparative analysis of UW-LightKey with cryptographic key generation approaches. As one can see, UW-LightKey shows the fastest generation speed (243 µs), as well as being the only approach in which the process of key generation is performed locally, without transmitting keys. Also, it uses only lightweight cryptographic operations (XOR, rotations, hash function).

TABLE 1 COMPUTATIONAL EFFICIENCY AND KEY TRANSMISSION SECURITY

Scheme	**Year**	**Core Operations**	**Key Generation Time**	**Key Transmission**	**Energy Efficiency**
Hash-based [7]	2022	Hash, XOR	N/A	Exchanged	High
ECC-based [8]	2023	ECC Multiplication	~ms	Exchanged	Medium
PUF/Chaotic [9]	2024	PUF, Chaotic maps	N/A	Exchanged	High
Lattice-based [10]	2025	NTRU	~ms	Exchanged	Medium
Lightweight AKA [15]	2024	Hash, XOR	~ms	Exchanged	High
UW-LightKey	2026	XOR, Rotation, ELH	243 µs	Computed locally	Very High

Table 2 provides a comparison of UW-LightKey and channel-based schemes. Unlike the current approaches, UW-LightKey utilizes deterministic physical parameters (acoustic delay and 3D distance estimation) in combination with node identity and shared salt. Such an approach allows the presented scheme to be the only channel-aware method enabling implicit mutual authentication, as it does not require complicated channel probing procedures. Overall, the above two tables demonstrate that UW-LightKey combines the strengths of two approaches at once – it has lightweight computation without any key transmission (as per Table 1) and takes advantage of the physical properties of the channel along with multiple entropy sources (as per Table 2).

TABLE 2 ENTROPY SOURCES AND PHYSICAL LAYER AWARENESS

Scheme	Year	Entropy Source	Uses Acoustic Channel?	Uses 3D Location?	Implicit Authentication?
Multi-hop Delay [11]	2023	Route propagation delay	✓	✗	✗
CIR Statistical [12]	2024	CIR distribution	✓	✗	✗
CIR Quantization [13]	2025	CIR + K-means	✓	✗	✗
Secure Localization [14]	2025	CIR + Neural Crypto	✓	✓	✗
UW-LightKey	2026	Acoustic delay + 3D distance + ID + Salt	✓	✓	✓

1) Research Motivation

The reasons for studying this topic can be found in the peculiarities of Underwater Wireless Sensor Networks (UWSNs). Firstly, due to the nature of the acoustic channel (large propagation delay and low data rate), the constraints imposed on the network protocols are strict, and efficient communication approaches should be used. Secondly, considering resource-limited end nodes, both energy-wise and computationally, the use of expensive cryptographic algorithms would be unreasonable. Thirdly, the vulnerability of acoustic channels to jamming and eavesdropping makes it important to have a reliable but lightweight cryptographic mechanism for ensuring security.

2) Research Gaps

However, despite the above reasons for researching the problem, there still exist several major gaps in this sphere. Firstly, there is a significant lack of balance between a high level of security and efficiency in many proposed approaches, leading to a trade-off between the two. There is a shortage of environmental context usage, as many of them miss the chance to take advantage of random channel parameters to generate secret information without using key exchange. There is a lack of flexibility in the protocols and their ability to work efficiently in large-scale environments, as well as a lack of evaluation regarding the most modern threats. Finally, there is a lack of a standardized evaluation procedure and criteria, limiting the comparison of the algorithms.

III. METHODOLGY

This section presents the systematic development methodology of the UW-LightKey key exchange protocol. The methodology is structured into four phases: (1) system model and assumptions, (2) lightweight cryptographic primitives, (3) the proposed protocol design, and (4) security threat model formulation.

A. Constraints and Requirements Identification

1) UWSN Environment Constraints

- High Propagation Delay: Acoustic propagation delay is significantly larger due to its lower speed (~1500 m/s) compared to radio-frequency (~3×10^8 m/s).
- Low Bandwidth: Low bandwidth in the acoustic channel requires minimizing communication overheads in protocol messages.
- Limited Energy: Replacement of batteries is not feasible due to frequent changes in location; therefore, optimization of energy costs of both computation and communication is necessary.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

- Low Power Processing Capabilities: Due to the use of small, energy-efficient, and cheap microcontrollers of 8-16-bit, computational capacity of nodes is limited.

2) Security Threat Model

- Security attacks that the new Lightweight UW Key Exchange algorithm should withstand include:
- Eavesdropping Attack: Intercepting communications between two parties passively.
- Man-in-the-Middle Attack: Intercepting and potentially tampering with communications actively.
- Replay Attack: Retransmitting intercepted protocol messages in order to cause malfunction or deception.
- Energy Drain Attack: Executing additional computations to exhaust the node energy capacity.

B. Selection of Lightweight Mathematical Foundation

The cryptographic base for our algorithm differs completely from that used in such classical cryptographic algorithms as RSA or Diffie-Hellman. Unlike other cryptosystems, UW-LightKey is based on a combination of cryptographic hashing functions, bitwise operations, and physical layer parameters as entropy sources.

Three categories of lightweight operations are employed in our algorithm:

- Bitwise XOR (⊕): An easily computed bitwise operation providing diffusion and confusion with relatively small computational costs.
- Bitwise Rotation (⋘): Cyclic shift operation increasing the amount of entropy.
- Cryptographic Hash Function (H): Collision and pre-image-resistant cryptographic hash ensuring message integrity with minimal overhead.

1) Enhanced Lightweight Hash (ELH) Function

To provide hashing functionality, we propose an Enhanced Lightweight Hash function (ELH). ELH is inspired by principles of SPARKLE [16] and PHOTON [17], but unlike the latter is optimized for software implementations (on microcontrollers). While SPARKLE and PHOTON focus on hardware implementation (on FPGA and ASIC), our ELH achieves high throughput and negligible latency when implemented in software on resource-constrained processors.

The proposed ELH hashing algorithm works with a 128-bit state, utilizing only three instructions: XOR, bitwise rotation, and addition modulo 256. The minimal number of instructions makes ELH highly efficient in computations on 8-16-bit microcontrollers commonly found in UWSN nodes.

The complete ELH algorithm is presented in Algorithm 1 (shown in Figure 1).

Algorithm 1: Enhanced Lightweight Hash (ELH) Function

```
Input:  Data string x (variable length)
        Pre-shared salt S (128-bit)
Output: Hash value h (128-bit)

// Initialize two 64-bit state words
h1 ← 0xABCDEF1234567890
h2 ← 0x0FEDCBA987654321

// Cryptographic constants
C1 ← 0x9e3779b97f4a7c15  // golden ratio (from SPARKLE)
C2 ← 0xbf58476d1ce4e5b9  // √2 - 1 (from PHOTON)
P ← 0x1FFFFFFFFFFFFFFF   // 61-bit prime modulus

// Process each byte of input and salt
for i = 1 to length(x) do
    byte ← x[i]
    salt_byte ← S[i mod length(S)]

    // Update first state word (inspired by SPARKLE)
    h1 ← ROTL(h1 + (byte × C1), 7)

    // Update second state word (inspired by PHOTON)
    h2 ← ROTR(h2 + (salt_byte × C2), 5)

    // Cross-mixing (dual-state diffusion)
    h1 ← h1 ⊕ h2
    h2 ← h2 + h1

    // Modular reduction
    h1 ← (h1 + C1) mod P
    h2 ← (h2 + C2) mod P
end for

// Combine both state words into 128-bit output
h ← (h1 ≪ 64) ⊕ h2
return h
```

Fig.1. Enhanced Lightweight Hash (ELH) Function.

2) Inspirations for Enhanced Lightweight Hash Design

SPARKLE: One of the main features of the SPARKLE algorithm is a dual-state cross-mixing scheme. In our work, we adopted the idea of alternately updating both states and XORing them, which is performed in (3).

$$h1 \leftarrow h1 \oplus h2,\ h2 \leftarrow h2 + h1 \tag{3}$$

Cross-mixing allows for achieving good diffusion since every single bit of input will influence the result of both state words after a few rounds.

PHOTON: The effectiveness of per-byte salt incorporation was successfully demonstrated by the authors of the PHOTON hash. Our hash function also implements this concept. In particular, every time a new byte is taken from the salt, it will affect the state values.

Enhanced Lightweight Hash Specification
This hash function takes a block of 128-bit state consisting of two words (h1, h2) with predefined initial values.

h1 = 0xABCDEF1234567890
h2 = 0x0FEDCBA987654321

For each input byte x[i] and salt byte S[i mod |S|], the following manipulations are performed:

h1 = ROTL(h1 + x[i] × C1, 7)
h2 = ROTR(h2 + S[i mod |S|] × C2, 5)
h1 ← h1 ⊕ h2
h2 ← h2 + h1
h1 = (h1 + C1) mod P
h2 = (h2 + C2) mod P

Where:
ROTL is the operation of bitwise rotating the input to the left.
ROTR is the operation of bitwise rotating the input to the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

right.
C1 = 0x9e3779b97f4a7c15; (Constant inspired by the golden ratio; related to SPARKLE)
C2 = 0xbf58476d1ce4e5b9; (Cryptographic constant; related to PHOTON)
P = 0x1FFFFFFFFFFFFFFF; (61-bit prime number).

3) Performance of ELH on CPUs

Rationale: While both SPARKLE and PHOTON have good cryptographic properties, they were designed to work efficiently mainly in hardware implementation (FPGA/ASIC). However, when working with software implementations, they demonstrate poor performance for three reasons. Firstly, S-box look-ups involve table look-ups (3–5 cycles) or bit-level operations. Secondly, matrix multiplication involves a multiplication over the Galois field 2^4 (5–6 cycles per cell in PHOTON's Mix Columns Serial). Thirdly, a larger state size is required; in particular, SPARKLE requires 256-bit states.

The following features of ELH design help to overcome the above-mentioned drawbacks:
(i) absence of S-boxes by using only operations XOR, rotations, and additions (one cycle each);
(ii) no matrix multiplication is needed; rather, only operations like modular addition are performed; and
(iii) the smaller state size (128-bit), fitting into two registers.

Benefits of ELH design for UW-LightKey. The advantages that come out of using ELH design are listed below:
(i) software-oriented design;
(ii) high performance; reaching 454 µs in time spent to generate the key, which is faster compared to SPARKLE/PHOTON in software implementation;
(iii) low energy consumption since the main operations consist of arithmetic operations;
(iv) endurance of security principles from SPARKLE and PHOTON;
(v) pre-shared salt preventing pre-computation attack;
(vi) avalanche effect – cross-mixing provides for high sensitivity to inputs; and
(vii) per-party nonces generation. Each party generates its nonce based on the initial value configuration.

C. Proposed Algorithm Design: UW-LightKey

1) Core Algorithm Components

The proposed algorithm is referred to as the Underwater Lightweight Key Exchange (UW-LightKey). The purpose of this algorithm is to provide a shared key exchange between two underwater nodes. For the task at hand, dynamic parameters of the physical layer channel are used in addition to predefined unique IDs of nodes: acoustic propagation delay τ and three-dimensional coordinates d. The general purpose of UW-LightKey is to provide computational lightness, energy efficiency, quick operation, and strong security due to the constraints characteristic of UWSNs.

Figure 2 shows the most important elements of the UW-LightKey design. The following three types of inputs are used in the UW-LightKey algorithm: (i) a unique 32-bit node ID; (ii) a 128-bit pre-shared salt, ensuring domain separation and immunity to the pre-computation attack; and (iii) channel measurements τ and d, which represent the entropy source for the current session. The design relies upon the usage of the ELH function (Algorithm 1), inspired by SPARKLE and PHOTON. As a result, the 128-bit session key is computed locally by each participant.

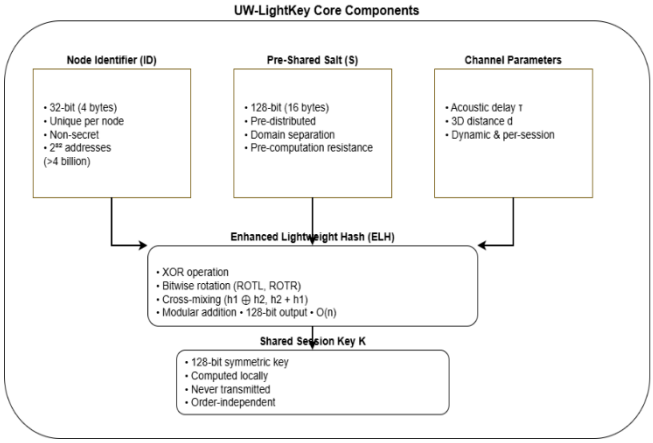

Fig.2. Core Algorithm Components.

UW-LightKey's core algorithm is based on three basic components, namely a pre-provisioned node identifier, a pre-shared secret salt, and dynamic measurements of the underwater acoustic channel parameters. These components are introduced in the following order:

1. Node Identifier (Node ID)

Every node is provided with a unique 4-byte (32-bit) node identifier. The identifier serves as a non-secret, long-lived node label used both for addressing and protocol execution.

Rationale Behind a 4-Byte Node Identifier:

- Sufficient Address Space: There exist 2^{32} (4 billion) distinct addresses in the 32-bit address space, which is sufficient even for the largest expected UWSN networks.

- Standard Protocol Support: 4 bytes is a common address size supported by all major network stacks (IPv4).

- Efficient Communication: The reduced length of the identifier decreases message size, thus improving bandwidth efficiency.

- Resistance to Attacks: A random 32-bit number represents a sufficiently large guessing space for ID-guessing attacks.

2. Pre-Shared Secret Salt

A 16-byte (128-bit) secret key is distributed in advance to all the legitimate nodes in the network. The salt serves as a domain-specific parameter that strengthens other secret cryptographic keys.

Rationale Behind a 128-Bit Secret Salt:

- Large Size: A secret key consisting of 128 bits presents an adequate security measure against brute-force attacks.

- Appropriate Computation Time: Computations over 128-bit numbers are performed efficiently on 8 to 16-bit microprocessors and do not require big-integer library support.

- Key Compatibility: The chosen secret size ensures the seamless use of the generated key in subsequent cryptographic algorithms (serves as the encryption key in ChaCha20).

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

3. Acoustic Channel Parameters

Two dynamic parameters related to the state of the underwater acoustic channel are used to generate the shared secret key. They serve as external entropy sources independent of the internal cryptographic engine.

- Shared Acoustic Signal Delay:

An additional parameter that corresponds to the measured delay of the acoustic signal sent between two communicating nodes in the network. In UW-LightKey, the shared acoustic delay is utilized as a mutually visible but difficult to estimate by a distant adversary dynamic secret seed.

Security Features:

- Spatial Fingerprinting: The delay acts as the fingerprint of each pair of nodes, uniquely determined by the exact distance between them, the local sound speed profile (which depends on salinity, temperature, and pressure), and the unique multi-path characteristics of the communication channel.
- Measurement Robustness: An eavesdropping party located remotely from the legitimate communicating nodes cannot measure the delay of the shared acoustic channel precisely. Ambient noise, interfering signals, and complex underwater acoustic phenomena make a remote measurement impossible.
- Temporal Fingerprinting: The delay between two nodes is subject to change because of their mobility (drifting caused by currents), the nature of the aquatic medium, and modifications to the effective acoustic path. Therefore, the generated cryptographic values will be unpredictable and can be used only once in each run.
- Physical Layer Authentication: The agreement on the mutual measurement of the acoustic delay proves cryptographically that the corresponding peer node exists in a particular physical environment compatible with the measured signal propagation time.
- Distance Between Nodes (d): In contrast to using raw geographical coordinates, the algorithm relies on computing Euclidean distance d to represent the geometric relation between nodes as a stable but distinguishing element. The use of d helps to abstract from a fixed coordinate system and incorporate the relative position of nodes directly in the key exchange.

4. Functions Used:

a. ELH() Function (Enhanced Lightweight Hash):

Description: To perform lightweight hashing at a sufficient speed and provide an acceptable security guarantee as described in Algorithm 1. It is used for generating nonces, calculating shared keys, and verifying message authenticity.

Function Characteristics:

- Length: 128 bits (16 bytes).
- Based on lightweight computations: XOR, rotation (ROTL, ROTR), cross mixing (h1 ⊕ h2, h2 + h1), modular addition.
- Salt: Domain separation is ensured through pre-shared salts used for each byte.
- Security features: Prevents an adversary from computing the same output (about 2^64 computational complexity).
- Motivation: Inspired by design ideas of SPARKLE and PHOTON protocols.

b. Generate_Acoustic_Nonce() Function:

Description: To generate a nonce, which is a function of the physical characteristics of the channel (acoustic delay, 3D coordinates of the nodes). A nonce is used in order to prevent replay attacks.

Function Work Steps:

- Measurement of Shared Acoustic Delay: Measurement of acoustic signal propagation time between two nodes.
- Measuring of Position: Obtaining the geographical coordinates of the node: (x,y,z).
- Generating of Nonce: Nonce = ELH((τ ⊕ d) + ID_other, S).

2) Complete Key Exchange Algorithm

The complete key exchange algorithm is presented in Algorithm 2 (shown in Figure 3).

Algorithm 2: UW-LightKey Key Exchange Algorithm (Online Phase)

```
Input: Node coordinates (x_A, y_A, z_A) and (x_B, y_B, z_B),
       shared acoustic delay τ, node IDs ID_A and ID_B,
       pre-shared salt S (128-bit)
Output: Shared session key K (128-bit)

for each communication session do
   Step 1: 3D Euclidean Distance Calculation
    d = sqrt ((x_B - x_A) ² + (y_B - y_A) ² + (z_B - z_A) ²)

   Step 2: Initial Nonce Construction
    Nonce_initial = (d ⊕ τ) + (ID_other ≪ 16)

   Step 3: Per-Party Nonce Generation
    for node A do
       Nonce_A = ELH ((τ ⊕ d) + (ID_B ≪ 16), S,128)
    end for

    for node B do
       Nonce_B = ELH ((τ ⊕ d) + (ID_A ≪ 16), S,128)
    end for

   Step 4: Shared Key Derivation
    K = ELH (Nonce_A ‖ Nonce_B,128) ⊕ ELH (Nonce_B ‖
Nonce_A,128)

   Step 5: Verify Key Agreement
    if K_A == K_B then
       return K
    else
       return error ("Key mismatch")
    end if
end for
```

Fig.3. UW-LightKey Key Exchange Algorithm (Online Phase).

3) Mathematical Components Used by the Proposed Algorithm

1. Three-dimensional Euclidean Distance Calculation:

Input: Coordinates of two nodes (x1, y1, z1) and (x2, y2, z2);

Calculation: The distance d is calculated using (1);

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (1)$$

Output: An integer value d;

Cryptographic Significance: The distance value obtained in this operation is linked to the physical environment. Hence, it serves as a basis for generating a unique key due to the intrinsic connection between the physical space and nodes.

2. Creation of Initial Nonce:

Inputs: Integer distance d, acoustic delay, and identifier of the node ID_other;

Computation: The initial nonce is created based on bitwise manipulations described by (2);

$$Nonce_{initial} = \left(d \oplus \text{shared}_{\text{delay}}\right) + (ID_{other} \ll 16) \quad (2)$$

Where: Operator $\oplus$ denotes bitwise XOR, while $\ll 16$ is used to represent a left shift operator by 16 bits;

Cryptographic significance: This operation provides a synchronized yet random seed to the parties, which depends on a variable parameter (shared acoustic delay) and a location-based one.

3. Lightweight Hash Function

The generated nonce is further passed through a lightweight hash function that performs computations on the limited computational resources available in embedded systems. Our proposition of the light hash function, named Enhanced Lightweight Hash (ELH), is based on ideas introduced in SPARKLE and PHOTON constructions. ELH is designed to run on software-based implementations of resource-constrained microcontrollers. The full algorithm description will be given in Algorithm 1 (Figure 1).

4. Nonce Computation and Party-to-Party Bindings:

Both nonces are computed separately from each other on the basis of the built initial value and channel parameters in ELH as defined in Algorithm 1. Asymmetric computations of parties are carried out to ensure the uniqueness while still synchronizing.

Node A: The nonce for Node A is generated by means of ELH as follows (see (3)):

$$Nonce_A = ELH\left((\tau \oplus d) + (ID_B \ll 16), S\right) \quad (3)$$

Node B: Similarly, for Node B, the nonce formula is (see (4)):

$$Nonce_B = ELH\left((\tau \oplus d) + (ID_A \ll 16), S\right) \quad (4)$$

Properties: The asymmetry of the two equations implies that Nonce_A and Nonce_B are generated using the same fundamental channel measurements and salt, but they are bound cryptographically to the ID of the second party. It leads to obtaining two distinct but still cryptographically related values, providing mutual input for the next stage of the key generation.

5. Shared Key Derivation**:**

At last, the shared session key K is generated by each party independently of their nonces, Nonce_A and Nonce_B. To make the key derivation invariant to any order and ensure maximum entropy mixing, the formula is used (see (5)).

$$K = ELH(Nonce_A \,||\, Nonce_B, S) \oplus ELH(Nonce_B \,||\, Nonce_A, S) \quad (5)$$

Where:

- operation ‖ corresponds to concatenation;
- function ELH(·, S) stands for ELH (described in Algorithm 1); and
- $\oplus$ stands for XOR.

Rationale and Security Considerations:

- Order Invariance: With XORing two values obtained by hashing both orders of concatenations, it is ensured that the order of participants' labeling will not influence the key computation process. That means that the resulting key will be obtained by both participants.

- High Entropy Mixing: The construction guarantees that all bits of both nonces affect the key generation after running the hash twice, thus increasing the entropy.

- Non-Transmitting of the Key: The symmetric key (128 bits) is generated locally by each participant without being transmitted over the channel.

4) Sequence Diagram

Figure 4 shows the sequence diagram of our proposed UW-LightKey key exchange algorithm, where the message flow between Node A and Node B is illustrated. At the beginning of the communication, each node will compute its spatial coordinates and the shared acoustic delay (τ). After completing the synchronization procedure, each node computes Euclidean distance d. Then, they use the ELH hash function to generate a unique nonce number. Both nodes calculate their own session key without transmitting any data over the network. Finally, after exchanging acknowledgements, the session key agreement phase is completed.

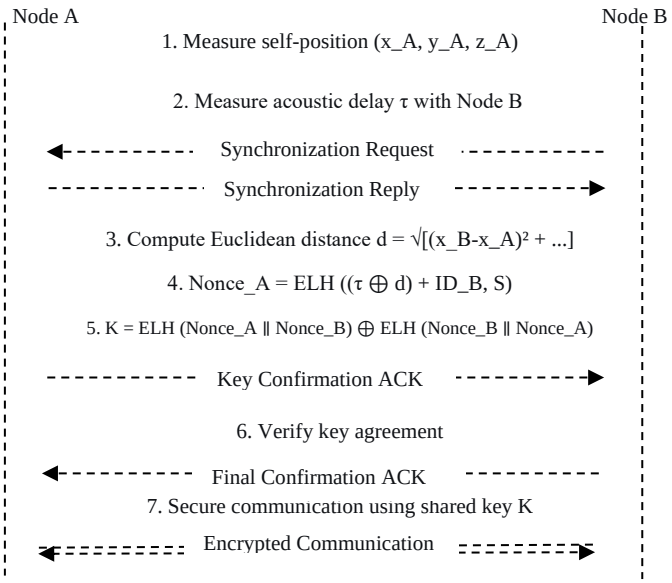

Fig 4. Sequence diagram of the UW-LightKey key exchange algorithm.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

5) Protocol Workflow

Figure 5 illustrates the flowchart of the entire process in the proposed UW-LightKey key-exchange protocol, presenting a series of operations carried out by two connected nodes denoted as nodes A and B. This diagram shows the application of measurements, cryptography, and final verification processes to generate a common key.

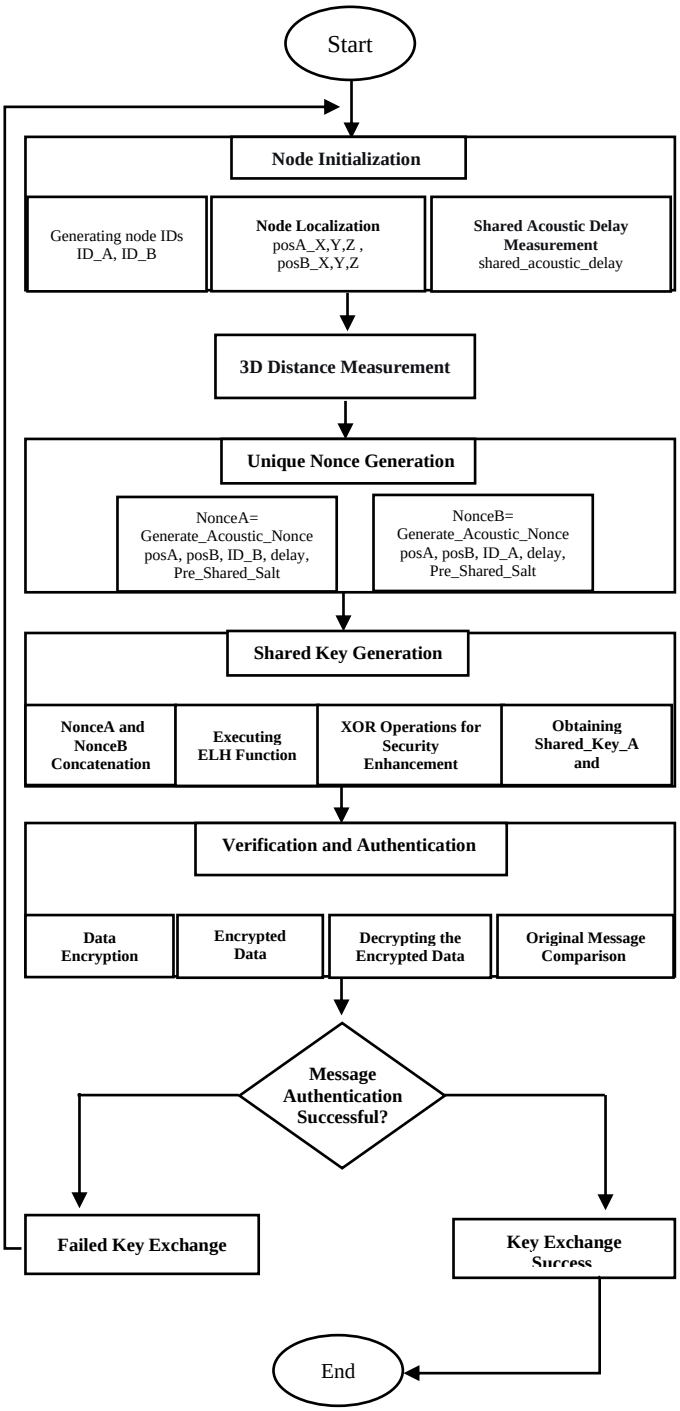

Fig. 5. Workflow of the UW-LightKey Key Exchange Algorithm

IV. SECURITY THREATS ANALYSIS AND COUNTERMEASURES IN UW-LIGHTKEY

A. Eavesdropping Threat

Threat: An outsider passively spies on the information transmitted between the two parties without interfering in any way.

Countermeasure:

- The data exchanged is encoded using a shared secret key based on stochastic environmental features.

- Communication is always encrypted to make data unreadable for an outsider who lacks the key.

B. Man-in-the-middle (MITM) Threat

Threat: A potential opponent tries to impersonate either one of the two parties involved to present a lie or a false message to them.

Countermeasure:

- Nonce Authentication: Authentication takes place against the Nonces generated according to environmental characteristics.

- Forgery Resistance: Any attempt to forge data would require forging a corresponding Nonce or a signature, which is impossible.

C. Replay Threats

Threat: A data packet is intercepted and reused to trick the counterpart about ongoing communication sessions or cause an unintended session reset.

Countermeasure:

- A new, different Nonce is generated each time.

- Verification of received messages by checking whether the Nonce has ever appeared before.

- Usage of environmental data makes replays extremely difficult.

D. Energy Depletion Threat

Threat: The energy supply of the target gadget is compromised by sending fake packets or flooding.

Countermeasure:

- A unique Nonce depending on real physical features makes such threats inoperable.

E. Cryptographic Mathematical Attacks

Threat: Adversaries try to break encryption systems using their complex calculations and knowledge of mathematical algorithms.

Countermeasure:

- Utilizing non-linear operations such as XOR, bit rotations, hashing, and concatenation of sources of randomness.

The Table 3 below represents the summary security analysis of the UW-LightKey algorithm.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

F. Quantum Security Evaluation

Given the increasing advancement in quantum computing, cryptographic protocols need to withstand emerging threats such as integer factorization and discrete logarithm quantum attacks via Shor's algorithm and quadratic speedup of brute-force attacks through Grover's algorithm. This section evaluates the quantum resistance of UW-LightKey to these threats, comparing post-quantum security features of UW-LightKey with classic cryptographic protocols.

TABLE 3 SECURITY ANALYSIS SUMMARY—UW-LIGHTKEY COUNTERMEASURES AGAINST UWSN ATTACKS

Attack	UW-Lightkey Protection Mechanism
Eavesdropping Attack	• The shared key is never transferred (calculated locally). • Cryptography based on unique physical channel characteristics. • Dynamic keys per session. • Pre-shared salt.
Man-in-the-Middle Attack	• Identifier bound to physical properties (position, delay). • Mutual verification via encryption/decryption. • Difficulty of simulating physical properties accurately.
Replay Attack	• Nonce values bound to single session. • Key linked to temporal parameters (acoustic delay). • Reliance on changing environmental conditions.
Energy Drain Attack	• Unique nonce based on physical properties prevents tampering and replay effectiveness.
Mathematical Cryptanalysis Attacks	• Non-linear operations (XOR, rotation). • Fragmentation and merging of random sources.

1) Resistance to Shor's Algorithm

The threat posed by Shor's algorithm to classic public-key cryptographic algorithms cannot be overstated. Shor's quantum algorithm solves problems related to prime factorization and hence breaks RSA. Similarly, quantum computing also poses risks to other public-key encryption algorithms like Diffie-Hellman and ECC that depend on the difficulty of the discrete logarithm problem [18].

UW-LightKey protocol stands out because it differs from conventional public-key cryptography schemes. First, it relies on entropy derived from the physics layer, namely, the acoustic delay τ and the three-dimensional distance d. These physical parameters are dynamic, and hence quantum computers do not provide any significant advantage for estimating them. Second, UW-LightKey uses XOR, bitwise rotations, and hashing operations for generating symmetric keys. As already noted, Grover's algorithm offers only a quadratic speedup in brute-forcing these keys, making quantum computers ineffective against this aspect of the UW-LightKey protocol. Third, the salt is symmetric, and quantum computing offers no significant advantage over Grover's quadratic brute-forcing algorithm. The key size of 128 bits translates to an effective key size of 2^{64}against Grover's attack, which is considered secure for the foreseeable future [18].

2) Resistance to Grover's Algorithm

Grover's quantum algorithm breaks all symmetric algorithms by shrinking the key size from 2^n to $2^{n/2}$. Since UW-LightKey uses 128-bit keys, its effective size becomes 2^{64}, which is considered quantum-resistant [19].

Moreover, the hash functions H(x, S) used in UW-LightKey are not immune to quantum collision attacks. BHT algorithm [20], which finds a collision with $O(2^{n/3})$ effort, can be used against a hash of size n bits in about $2^{n/3}$ time. Using this approach, a 128-bit hash requires roughly $2^{42.7}$steps. However, dynamic parameters used in UW-LightKey (τ, d) reset the cryptographic parameters after every session and therefore mitigate any offline quantum attacks.

3) Comparison of UW-LightKey to Post-Quantum Algorithms

With recent advances in the post-quantum field, some of the popular encryption algorithms include CRYSTALS-Kyber and NTRU algorithms based on lattices and McEliece based on code, as well as hash functions such as SPHINCS+ [21]. These algorithms offer quantum resistance at the expense of heavy computations. Therefore, they are unsuitable for use in UWSNs.

Table 4 gives a comparison between UW-LightKey and some of the most famous quantum-resistant algorithms.

TABLE 4 QUANTUM SECURITY COMPARISON OF CRYPTOGRAPHIC SCHEMES

Scheme	Cryptographic Basic	Vulnerable to Shor?	Vulnerable to Grover?	Suitability for UWSNs
RSA	Integer Factorization	Yes (broken)	N/A	Low (high overhead)
ECC	Discrete Logarithm	Yes (broken)	N/A	Medium
Diffie-Hellman	Discrete Logarithm	Yes (broken)	N/A	Low
CRYSTALS-Kyber	Lattice Problems	No	N/A	Low (high overhead)
UW-LightKey	Physical Layer + Symmetric	No	**2^{64} security**	High (lightweight)

V. PERFORMANCE EVALUATION AND RESULTS

In this section, we provide a thorough performance analysis of the proposed UW-LightKey algorithm by benchmarking its efficiency against existing cryptographic solutions (RSA, ECC, and DH). We analyze whether the algorithm meets the challenges imposed by resource-constrained environments and discuss the results based on the following performance metrics:

- Execution Speed: Estimated through key generation time and throughput.

- Efficiency: Evaluated by measuring the number of basic operations required to generate one bit of the key and the overall operation count.

- Memory Consumption: Calculated by analyzing the amount of dynamic memory used and corresponding power consumption during computations.

A. Experimental Setup

A dedicated performance testing tool implemented using the C++ language and Visual Studio 2024 was used in order to emulate typical underwater sensor node conditions and assess the efficiency of each algorithm implementation accordingly.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

The following performance analysis parameters were taken into account:

- Statistical Significance: Each algorithm was run 1,000 times in a cycle in order to make sure that the results are statistically reliable and free from system noise and random variations.
- Execution Time: Key-generation time was estimated using microseconds as a unit of measure by employing a high-resolution timer like std::chrono::high_resolution_clock.
- Memory Allocation: Working set size was monitored using the Windows API function GetProcessMemoryInfo in order to measure memory allocation in bytes.
- Operation Count: The custom software counters were included in each cryptographic algorithm implementation to determine the number of basic operations performed (modular multiplications, hashing iterations, bitwise operations).

Input Data:

The UW-LightKey algorithm was given randomized yet realistic input data per iteration (including 3D coordinates for the distance calculation and random delay value within the 100-500 ms range). Other algorithms received randomized, cryptographically secure random seeds for their key generation process.

B. Performance Metrics and Evaluation

In order to offer both quantitative and comparative analysis, the following performance criteria shall be introduced. Specifically, the criteria were chosen such that they reflect the efficiency and appropriateness of implementing the considered key exchange algorithm in a resource-constrained environment typical for the UWSNs.

1. Total Average Time of Key Generation (T_avg):

Definition: The arithmetic mean of the time it takes to execute the entire key generation routine once, from receiving the input parameters to obtaining the final shared secret.

Measure/Unit: Determined by calculating the average of times measured throughout all 1,000 iterations and reported in microseconds (μs).

Importance: T_avg represents the principal characteristic of algorithm efficiency. The smaller it is, the faster key agreement sessions could be conducted, thus allowing to reduce the processor load.

2. Throughput (Θ):

Definition: The average rate at which cryptographic keys are generated.

Calculation: Θ = 1,000,000/T_avg (in μs); reported in keys per second.

Importance: Defines the processing capacity of the algorithm. The higher it is, the better.

3. Peak RAM Memory Allocation

Definition: The maximum amount of volatile memory allocated during the operation of the key generation algorithm.

Measurement: Obtained through the usage of system-level profiling tools (GetProcessMemoryInfo function on Windows). The metric is reported in kilobytes (KB).

Importance: It defines how suitable the algorithm is for implementation in microcontrollers with a relatively small amount of available RAM. The smaller it is, the more resources will remain for application-specific processing.

4. Computational Operations per Bit (Ops/bit):

Definition: The average amount of simple arithmetic/logical operations (modular additions/multiplications, bitwise XOR rotations, hash function computations) required to obtain one bit of the final shared 128-bit key.

Calculation: Total Operation Count / (128 bits * Number of Iterations).

Importance: Provides an intrinsic characteristic of algorithm compactness, thus reflecting its elegance. The fewer operations required per obtained bit, the more efficiently the key generation process operates; therefore, it consumes less energy. An important property, as energy efficiency is critical in battery-powered environments.

5. Total Operation Count:

Definition: The total number of counted simple operations performed during the entire process of generating 1,000 keys.

Importance: Provides the gross computational complexity measure, which helps to better understand the ops/bit metric and comprehend the total computational workload imposed on the processor.

The methodology used for the evaluation of the efficiency of the cryptographic key exchange algorithms is provided in Table 5. The five main metrics have been highlighted, with an explanation of their computation and importance in the context of constrained environments.

TABLE 5 PERFORMANCE EVALUATION METRICS

Metric	Description	Significance	Formulation
Determines algorithm speed in key generation	$\frac{\text{Total Time}}{\text{Number of Iterations}}$	Time required to generate a single key	Average Key Generation
Measures the algorithm's production capacity	$\frac{1{,}000{,}000}{T_{avg}(\mu s)}$	Number of keys generated per second	Throughput
Demonstrates resource utilization efficiency	Using system tools (e.g., GetProcessMemoryInfo)	Amount of memory used during execution	Memory Usage
Measures computational complexity relative to key length	$\frac{\text{Total Operations}}{\text{Key Length} \times \text{Number of Iterations}}$	Average operations required to produce each key bit	Operations per Bit
Shows the overall processor load and energy consumption	$\sum_{i=1}^{n} ops_i$	Total number of computational operations across all iterations	Total Operations

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

C. Result

1) Key Generation Time:

According to the results in Figure 6, the algorithm UW-LightKey takes the shortest time for generating keys, 243 microseconds. The reason for this performance can be found in the fact that the algorithm was designed specifically for the aquatic environment. Thus, it utilizes simple math calculations as well as physical characteristics of the acoustic communication channel, i.e., distance and delay. Traditional algorithms take much more time to generate keys due to complicated mathematical calculations. Specifically, ECC is the second-best solution (926 microseconds), followed by DH (2152 microseconds). Finally, RSA performs the worst in terms of generating keys (3633 microseconds).

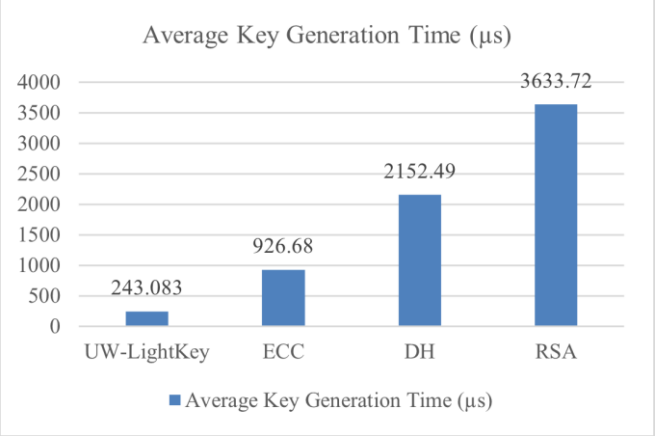

Fig.6. Average Key Generation Time (μs).

2) Throughput:

Results in Figure 7 of throughput show the clear superiority of the UW-LightKey scheme, which generates keys at a rate of 4114 keys/second. This is made possible through the efficient key generation approach, characterized by quick mathematical calculations, as well as the utilization of environmental characteristics. In comparison, traditional schemes have throughput showing a declining trend: ECC, which uses moderately quick elliptic curve operations, manages only 1079 keys/second; DH comes next, with 465 keys/second because of its computationally intensive logarithm calculations; and finally, RSA manages a meager 275 keys/second owing to the heavy calculation of big number arithmetic operations. This demonstrates that UW-LightKey is indeed the best scheme where high throughput is needed, like in underwater sensor networks.

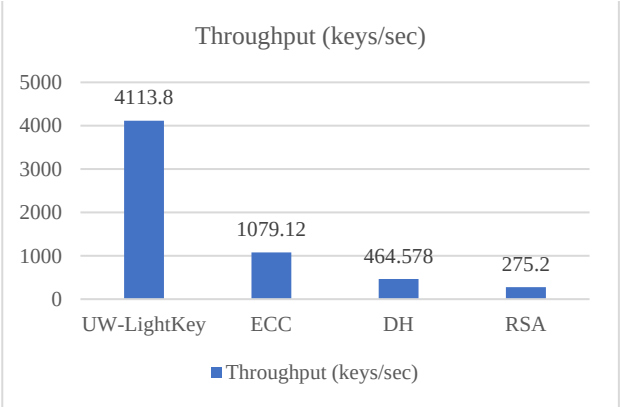

Fig. 7. Throughput (keys/sec)

3) Memory Usage:

Based on the outcome in Figure 8, it is evident that UW-LightKey demonstrates significant superiority regarding memory utilization, as this protocol utilizes the least memory compared to the others tested, measured in kilobytes. The efficiency associated with memory consumption emanates from the fact that it is designed using light hash functions along with basic mathematics. In contrast, conventional algorithms have shown an increasing trend in memory utilization, where ECC uses 5104 KB for storing elliptic curve data; the DH algorithm consumes 5292 KB due to storing large primes; while RSA demands the largest amount of memory of 5700 KB due to high computation and very large primes. Therefore, the most suitable protocol would be UW-LightKey.

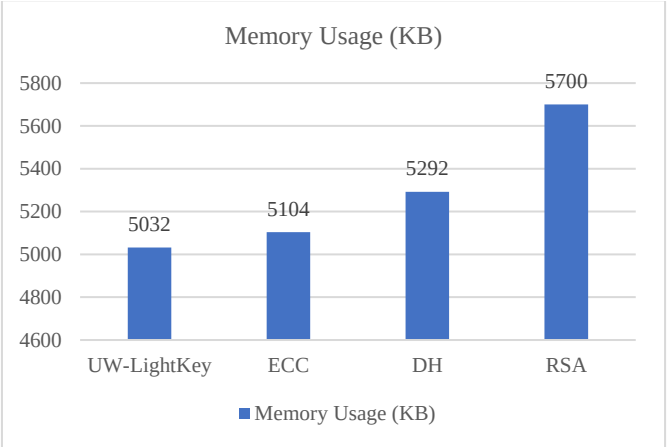

Fig.8. Memory Usage (kb)

4) Operations per Bit:

In Figure 9, the data on operations per bit reveal an obvious superiority of the UW-LightKey algorithm that features the lowest number of operations per bit. The reason is the use of simple arithmetic operations, including hashing, XOR, and bit rotations, that are not complicated calculations. Meanwhile, the traditional cryptographic algorithms have a high number of operations per bit since ECC has an average number of operations because of elliptic curve multiplications; DH has a high number due to the large number of exponentiations in prime fields; while RSA has the highest number of operations per bit due to large integers factorization and the number of exponentiations needed for encryption.

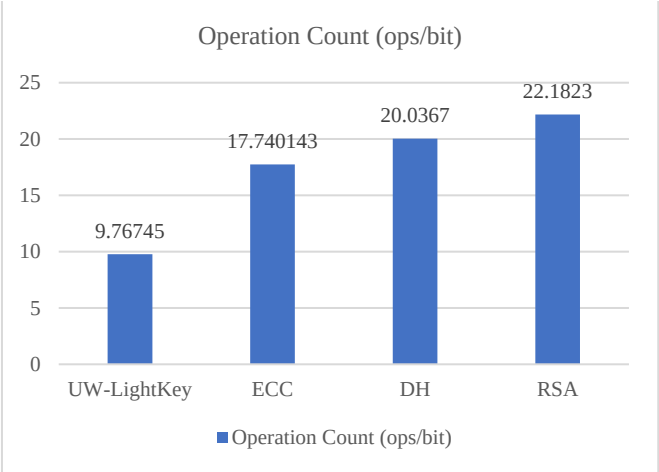

Fig.9. Operation Count (ops/bit).

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

5) Total Operations:

About total operations, Figure 10 shows that the performance of UW-LightKey is superior due to its low total operation count, as the lowest total operation count is observed in the case of UW-LightKey. The reason behind this lies in the algorithm design itself, whereby it avoids the complex computational process in traditional algorithms by using simple processes like hashing and XORing to avoid the need for heavy exponentiations or other calculations. On the other hand, traditional algorithms see an increase in operation counts gradually, where ECC has a moderate total operation count due to elliptic curve arithmetic, DH involves logarithms, and RSA uses maximum operation counts since a lot of mathematics is involved in prime factorization.

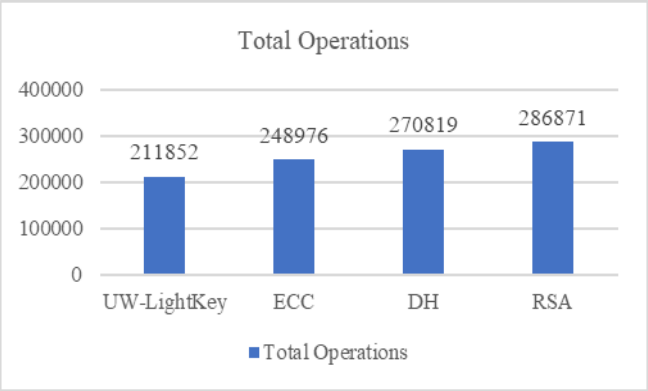

Fig.10. Total Operations.

These results show that the most efficient algorithm is the UW-LightKey algorithm because of its low computational complexity.

In Table 6, four cryptographic algorithms are compared, and the UW-LightKey algorithm is shown to be an ideal candidate for encryption for aquatic devices due to its lightweight structure created using hashing and stream ciphering. The lightweightness allows us to say that the UW-LightKey has an excellent performance level, including very low computational complexity, low energy usage, and minimal memory consumption. Meanwhile, ECC gives the best security performance due to the utilization of elliptic curves while providing good performance as well. RSA and DH are the least efficient since they require a lot of energy and memory as they work with very complex mathematical operations.

TABLE 6 COMPARATIVE ANALYSIS OF CRYPTOGRAPHIC ALGORITHMS—UW-LIGHTKEY VS. ECC, RSA, AND DH

Algorithms Factors	Diffie-Hellman Algorithm	RSA Algorithm	ECC Algorithm	UW-LightKey Algorithm
Cipher Type	Asymmetric	Asymmetric	Asymmetric	**Lightweight Hash-based**
Cryptographic Family	Discrete Logarithm Problem (DLP)	Integer Factorization Problem (IFP)	Elliptic Curve Cryptography	Hash + XOR Stream Cipher
Complexity	Medium	High	Medium	Very Low
Energy Consumption	High	Very High	Low	Very Low
Speed	Slower than ECC	Slowest	Faster than RSA and DH	Fastest
Memory Usage	Moderate	Highest	Relatively Low	Lowest
Primary Purpose	Key Exchange Only	Key Exchange, Digital Signatures, and Encryption	Key Exchange and Digital Signatures	Key Exchange (specialized for underwater environments)

VI. CONCLUSION

The UW-LightKey algorithm represents a novel, resource-efficient approach for exchanging keys designed for Underwater Wireless Sensor Networks (UWSNs). Using the characteristics of acoustic signal propagation through the water, such as delay and the three-dimensional spatial position of sensors, along with simple mathematical calculations, the algorithm provides efficient security in a resource-constrained environment. The experimental results have clearly demonstrated the superiority of the proposed approach over traditional key exchange methods, including ECC, RSA, and Diffie-Hellman algorithms, in terms of such important parameters as key generation speed, memory and energy consumption, as well as computational cost. Additionally, the presented algorithm appears to be resistant to multiple kinds of security vulnerabilities typically associated with UWSN operation.

A. Future Directions

To further enhance the robustness and applicability of the UW-LightKey protocol, the following research directions will be explored:

1. Advanced Network Simulation and Field Testing: Testing the protocol in special underwater simulators (e.g., Aqua-Sim) to measure network-layer performance (such as packet delivery ratio and throughput) both in the presence of acoustic noise and random node motion (realistic conditions), then experiment at sea.
2. Independent Self-Tuning Capability: Create a system that can tune protocol parameters adaptively according to changes in the acoustic channel (temperature, underwater currents, and Doppler effect) as we want our MODs to be synchronized independently with different aquatic mediums.
3. Cross-Layer Routing Integration: Explore integration points of UW-LightKey with existing underwater routing protocols in order to support an end-to-end security solution while preserving the overall energy and computation efficiency within the network.
4. Formal verification and security hardening, e.g. using formal verification tools (e.g., Scyther) in order to obtain a mathematical proof of security against advanced attackers, along with lightweight post-quantum cryptographic primitives.
5. Scalability and Group Key Management: Designing scalable versions of the algorithm specificallyfor large-scale, high-density networks, including new Group Key Management architectures for cluster-based UWSNs.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

6. Decentralized Initialization: Developing decentralized protocols for dynamic initial-trust establishment and salting to replace the need for pre-shared keys (PPKs) or static salts.

REFERANCES

[1] F. P. F. Domingos, A. Lotfi, I. K. Ihianle, O. Kaiwartya, and P. Machado, "Underwater Communication Systems and Their Impact on Aquatic Life—A Survey," *Electronics*, vol. 14, no. 1, pp. 1-30, 2024. DOI: 10.3390/electronics14010007.

[2] K. Saeed, W. Khalil, A. S. Al-Shamayleh, S. Ahmed, A. Akhunzada, S. Z. Alharthi, and A. Gani, "A Comprehensive Analysis of Security-Based Schemes in Underwater Wireless Sensor Networks," *Sustainability*, vol. 15, no. 9, pp. 1-27, 2023. DOI: 10.3390/su15097198.

[3] S. U. Khan, Z. U. Khan, M. Alkhowaiter, J. Khan, and S. Ullah, "Energy-Efficient Routing Protocols for UWSNs: A Comprehensive Review of Taxonomy, Challenges, Opportunities, Future Research Directions, and Machine Learning Perspectives," *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 7, pp. 1-23, 2024. DOI: 10.1016/j.jksuci.2024.102128.

[4] P. Konduru and N. P. Nethravathi, "Secure and Energy-Efficient Routing Protocol Based on Micro-Segmentation and Batch Authentication," *Computer Networks*, vol. 248, pp. 1-14, 2024. DOI: 10.1016/j.comnet.2024.110293.

[5] P. S. Suryateja and K. V. Rao, "A Survey on Lightweight Cryptographic Algorithms in IoT," *Cybernetics and Information Technologies*, vol. 24, no. 1, pp. 21-34, 2024. DOI: 10.2478/cait-2024-0002.

[6] N. Zahra, S. Khawatmi, and B. A. Kassab, "A Comprehensive Survey of Security Challenges and Modern Solutions in Underwater Wireless Sensor Networks," *International Journal of Computer Applications*, vol. 187, no. 32, pp. 24-33, Aug. 2025. DOI: 10.5120/ijca2025925570.

[7] C. M. Kumar, R. Amin, and M. Brindha, "SafeCom: Robust Mutual Authentication and Session Key Sharing Protocol for Underwater Wireless Sensor Networks," *Journal of Systems Architecture*, vol. 130, pp. 1-9, 2022. DOI: 10.1016/j.sysarc.2022.102650.

[8] S. Shah, A. Munir, A. Waheed, A. Alabrah, M. Mukred, F. Amin, and A. Salam, "Enhancing Security and Efficiency in Underwater Wireless Sensor Networks: A Lightweight Key Management Framework," *Symmetry*, vol. 15, no. 8, pp. 1-15, 2023. DOI: 10.3390/sym15081484.

[9] Q. Xie and Y. Yao, "PUF and Chaotic Map-Based Authentication Protocol for Underwater Acoustic Networks," *Applied Sciences*, vol. 14, no. 13, pp. 1-15, 2024. DOI: 10.3390/app14135400.

[10] F. Jiang and M. Xu, "Security Authentication Protocol for Underwater Sensor Networks Based on NTRU," *Journal of Marine Science and Engineering*, vol. 13, no. 4, pp. 1-21, 2025. DOI: 10.3390/jmse13040742.

[11] R. Diamant, S. Tomasin, F. Ardizzon, D. Eccher, and P. Casari, "Secret Key Generation from Route Propagation Delays for Underwater Acoustic Networks," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 3318-3333, 2023. DOI: 10.1109/TIFS.2023.3280040.

[12] R. Diamant, P. Casari, F. Ardizzon, S. Tomasin, T. Corne, and B. Sherlock, "A Key Agreement Algorithm for Securing Underwater Acoustic Communications," *in OCEANS 2024*, pp. 1-5, 2024. DOI: 10.1109/OCEANS51537.2024.10682306.

[13] R. Diamant, P. Casari, F. Ardizzon, S. Tomasin, B. Sherlock, and T. Corne, "Channel-Based Key Generation for Secure Underwater Acoustic Communications," *IEEE Transactions on Wireless Communications*, vol. 24, no. 7, pp. 5678-5693, 2025. DOI: 10.1109/TWC.2025.3548670.

[14] R. Fan, A. Boukerche, P. Pan, Z. Jin, and Y. Su, "An Underwater Secure Localization Scheme Based on Physical Layer Cryptographic Learning," *IEEE Transactions on Mobile Computing*, vol. 25, no. 1, pp. 37-53, 2026. DOI: 10.1109/TMC.2025.3591016.

[15] A. M. Almuhaideb and D. M. Al-Khulaifi, "An Efficient Authentication and Key Agreement Scheme for the Internet of Underwater Things (IoUT) Environment," *IEEE Access*, vol. 12, pp. 175773-175789, 2024. DOI: 10.1109/ACCESS.2024.3504602.

[16] C. Beierle, A. Biryukov, L. Cardoso dos Santos, J. Großschädl, L. Perrin, A. Udovenko, V. Velichkov and Q. Wang, "Lightweight AEAD and Hashing using the Sparkle Permutation Family," *IACR Transactions on Symmetric Cryptology*, vol. 2020, no. 1, pp. 208-261, 2020. DOI: 10.13154/tosc.v2020.iS1.208-261.

[17] J. Guo, T. Peyrin, and A. Poschmann, "The PHOTON Family of Lightweight Hash Functions," *in Advances in Cryptology – CRYPTO 2011*, pp. 222-239, 2011. DOI: 10.1007/978-3-642-22792-9_13.

[18] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484-1509, 1997. DOI: 10.1137/S0097539795293172.

[19] L. K. Grover, "A Fast Quantum Mechanical Algorithm for Database Search," *in Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 212-219, 1996. DOI: 10.1145/237814.237866.

[20] Z. Zhang, W. Wu, and J. Zou, "Quantum Differential Collision Key Recovery Attack of Multi-Round EM Structure," *in Journal of Computer Research and Development*, vol. 58, no. 12, pp. 2811-2818, Dec. 2021. DOI: 10.7544/issn1000-1239.2021.20200427.

[21] P. R. Babu, S.A. Kumar, A. G. Reddy, and A. K. Das, "Quantum Secure Authentication and Key Agreement Protocols for Iot-Enabled Applications: A Comprehensive Survey and Open Challenges," *in Computer Science Review*, vol. 54, pp. 1-31, Nov. 2024. DOI: 10.1016/j.cosrev.2024.100676.