Journal of *Marine Science and Engineering*

MDPI

Article

A Blockchain-Based Edge Computing Group Signature Authentication Model for Underwater Clustered Networks

Yanxia Chen [1], Zhe Li [2] and Rongxin Zhu [2,*]

[1] School of Cyberspace Security, Hainan University, Haikou 570228, China; yxchen@hainanu.edu.cn
[2] School of Computer Science and Technology, Hainan University, Haikou 570228, China; lizhe@hainanu.edu.cn
* Correspondence: rxzhu@ieee.org

Abstract: Underwater Wireless Sensor Networks (UWSNs) are pivotal for advancing maritime capabilities. These networks predominantly utilize acoustic communication, characterized by an open and shared acoustic channel and energy-limited underwater nodes, which underscores the critical importance of node authentication and management. Blockchain technology, recognized for its security, confidentiality, and traceability, is particularly suitable for scenarios requiring secure data exchange. This paper proposes a blockchain-based collaborative node authentication model tailored for clustered networks in UWSNs to tackle the challenges posed by the open nature of acoustic channels and the constrained energy resources of underwater nodes. Autonomous Underwater Vehicles (AUVs) are deployed as blockchain nodes to aid cluster heads in identity verification, while all underwater acoustic nodes are integrated as lightweight blockchain nodes, thus ensuring uniform management and authentication. Furthermore, this study enhances existing clustering algorithms to prolong the operational lifespan of the network and introduces a group signature and authentication mechanism tailored to the unique conditions of underwater blockchain edge computing. This mechanism includes a robust two-round block verification scheme designed to secure the blockchain against potential consensus algorithm attacks. Comprehensive simulations are presented, validating the effectiveness of the proposed group signature solution in enhancing the security and sustainability of underwater clustered networks.

Keywords: underwater acoustic sensor networks; blockchain; clustering

check for updates

Academic Editor: Graciliano Nicolás Marichal

Received: 9 December 2024
Revised: 25 December 2024
Accepted: 27 December 2024
Published: 28 December 2024

Citation: Chen, Y.; Li, Z.; Zhu, R. A Blockchain-Based Edge Computing Group Signature Authentication Model for Underwater Clustered Networks. *J. Mar. Sci. Eng.* **2025**, *13*, 27. https://doi.org/10.3390/jmse13010027

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Underwater Wireless Sensor Networks (UWSNs) represent a significant extension of Wireless Sensor Networks (WSNs), which are dedicated to applications in marine environments. These networks consist of various nodes including sensor nodes, sink nodes, and mobile collectors, coupled with surface base stations [1–3]. UWSNs are pivotal in a range of applications from marine environment monitoring and offshore exploration to navigation assistance and tsunami warning. Furthermore, in the realm of national security, they play a critical role in the detection of underwater threats such as torpedoes and submarines.

Despite their vast potential, UWSNs face numerous challenges that hinder their widespread deployment. Key among these are issues related to node reliability and the scalability of network architectures [4]. The sensor nodes, deployed across diverse underwater locales, gather critical data on parameters like salinity and temperature, transmitting this information to sink nodes. These data are subsequently relayed to surface or land-based stations. Mobile collectors, such as Autonomous Underwater Vehicles (AUVs) and

underwater gliders, follow predetermined routes to collect data from these sensors [5–7]. The topology of these networks is depicted in Figure 1.

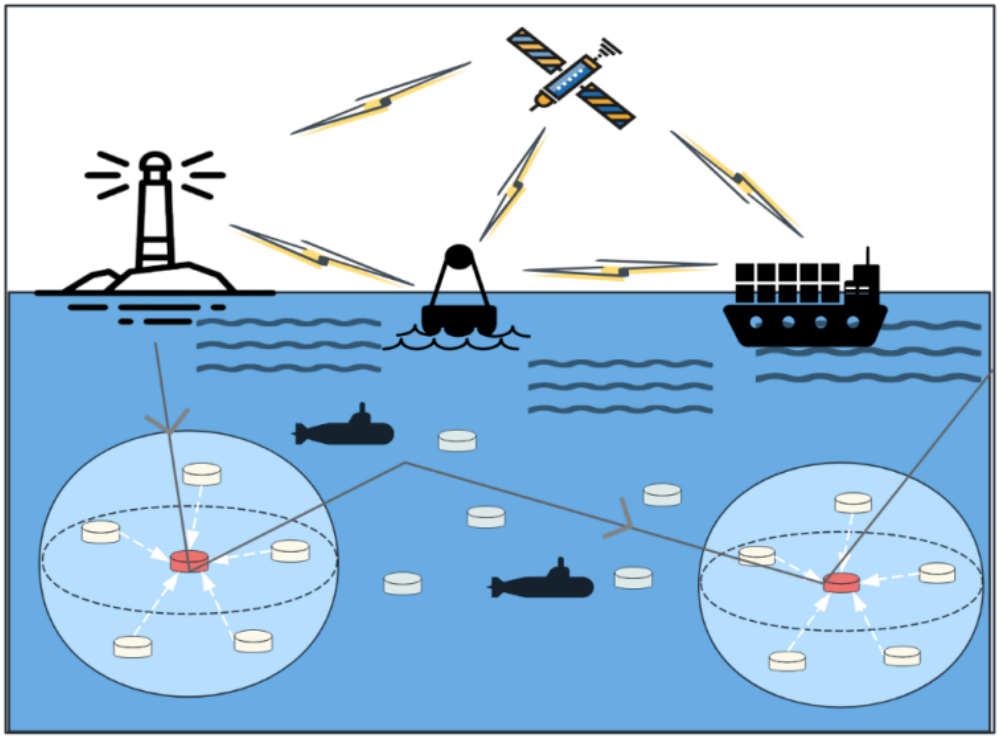


Figure 1. Topology of an underwater acoustic sensor network.

Underwater communication primarily utilizes acoustic signaling, which is capable of spanning distances up to 20 km or more. However, this mode of communication is fraught with challenges such as complex channel conditions and significant delays. Moreover, the openness and shared nature of the underwater acoustic channel exacerbate the difficulty in managing node authentication, making it a critical area of concern [8–10].

The harsh underwater environment poses further difficulties, primarily energy limitations and challenges in identity authentication management [11]. Nodes are often randomly distributed with restricted energy supplies, complicating uniform management and impacting the network's operational capacity and stability.

To address these critical issues, this paper proposes a novel blockchain-based cooperative node authentication model. This model leverages blockchain technology to manage underwater nodes by clustering them and using AUVs as blockchain nodes to assist cluster heads in identity verification. The decentralization, immutability, and traceability features of blockchain enhance the security of node identity authentication across the cluster.

The principal contributions of this investigation can be summarized as follows:

- Implements a blockchain framework to manage underwater nodes by clustering, utilizing AUVs as blockchain nodes for enhanced security in identity verification.
- Introduces an innovative clustering algorithm specifically designed to minimize energy consumption and extend the operational lifespan of networks under the unique constraints of underwater environments.
- Develops a group signature and authentication mechanism based on blockchain technology, featuring a two-round block verification process to secure node communications against consensus attacks.
- Conducts security analyses to validate the effectiveness of the proposed group signature scheme, demonstrating its resilience against various security threats in underwater settings.

The organization of the paper is outlined as follows: Section 2 provides a thorough review of the relevant literature, setting the stage for the subsequent discussions. Section 3 describes the network architecture foundational to this study, explaining its key components and functionalities. The methodology employed in our proposed model is elaborated in Section 4. Section 5 examines the security features and analyzes the algorithmic complexity of the proposed model. Performance evaluations, conducted through extensive simulations, are presented in Section 6. Finally, Section 7 concludes the paper, summarizing the major findings and contributions of our research.

2. Related Work

2.1. Clustering Algorithms for Underwater Wireless Sensor Networks

Using clustering strategies to manage the data transmission of underwater nodes is one of the key approaches to addressing the energy consumption challenges in UWSNs [12,13]. By employing clustering algorithms to aggregate and transmit data in underwater networks, reasonable clustering can alleviate the energy consumption and signal interference associated with long-distance data transmission, thereby improving the overall network efficiency. This approach increases the overall energy utilization of the network, and combining clustering algorithms with load-balancing strategies can significantly extend the network's lifetime.

To enhance the energy efficiency of UWSNs, researchers have deeply explored and improved numerous clustering strategies. Subramani et al. [14] proposed a metaheuristic algorithm for data clustering and routing decision making, which effectively selects cluster head nodes and determines the best routing paths. Domingo et al. [15] introduced an improved clustering algorithm, DUCS, which minimizes data loss and maintains communication quality by utilizing non-flooding routing techniques, proactive routing, and dynamic time synchronization. Sun et al. [16] proposed a low-power, low-latency LEACH protocol, which optimizes the cluster head election mechanism, communication cost functions, and data-forwarding functions, effectively addressing the energy consumption and data transmission delay issues of the LEACH protocol.

Tian et al. proposed a centralized control-based clustering scheme [17] to address energy optimization issues in underwater acoustic sensor networks. By employing node density-adaptive clustering, establishing in-cluster controllers, and optimizing relay node and relay cluster selection, the scheme achieves global energy balance optimization, effectively extending the network lifetime. Abdulzahrad et al. introduced an energy-efficient fuzzy unequal clustering and sleep scheduling protocol [18], which extends the network lifetime and reduces energy consumption through clustering, scheduling, and data transmission optimization. This method uses fuzzy C-means to form unequal clusters to balance energy consumption and selects cluster heads using a fuzzy logic system. It also employs a sleep scheduling strategy to reduce the number of transmitting nodes, thus improving network residual energy and lifespan.

In [4], a novel energy-efficient data collection scheme based on AUVs was proposed. This scheme utilizes a new cluster head selection method and transmission strategy suitable for large-scale AUV-assisted clustering in UWSNs. The scheme effectively reduces collection delays by eliminating AUV wait times within each cluster. Shah et al. [1] introduced an energy-optimized cluster head selection method based on an improved Tuna Fish Optimization Algorithm (ECERO) for UWSNs. This method considers energy, Euclidean distance, node density, average network energy, acoustic path loss models, and adaptive cluster head numbers while incorporating sleep scheduling for in-cluster nodes to reduce cluster head node load. This improves the performance of the EOCSR algorithm and effectively resolves hotspot issues caused by data relaying.

In [19], Shah et al. proposed a cluster-based cooperative energy-efficient routing (CEER) protocol, which solves energy consumption problems in traditional underwater node collaboration mechanisms and the LEACH protocol in underwater acoustic sensor networks. By optimizing clustering and cooperation strategies, CEER achieves load balancing and significantly improves network performance.

Table 1 summarizes the key clustering algorithms, their features, and their contributions.

Table 1. Summary of algorithms and their contributions in underwater network optimization.

Algorithm	Key Feature	Contributions	Refs
Metaheuristic Algorithm	Cluster head selection and routing optimization	Efficiently selects cluster heads and determines optimal routing paths for data transmission	[14]
DUCS	Non-flooding routing, proactive routing, dynamic synchronization	Minimizes data loss and maintains communication quality in underwater networks	[15]
LEACH protocol optimization	LEACH Protocol Optimization	Optimizes cluster head election and communication cost functions to address energy and delay issues	[16]
Central Control-based Scheme	Node density-adaptive clustering and relay optimization	Extends network lifetime by achieving global energy balance optimization	[17]
Fuzzy Unequal Clustering	Unequal clustering based on fuzzy logic	Optimizes cluster head selection and incorporates sleep scheduling	[18]
AUV-Assisted Data Collection	Autonomous Underwater Vehicle (AUV) assistance	Reduces collection delay and optimizes transmission strategies for AUV-assisted UWSNs	[4]
ECERO	Skipjack Tuna Optimization Algorithm (ECERO)	Optimizes energy consumption and solves the hotspot issue with energy-efficient cluster head selection	[1]
CEER	Cooperative energy-efficient routing (CEER)	Improves network performance by optimizing clustering and cooperation strategies	[19]

2.2. Blockchain Consensus Security

In the underwater network environment, challenges related to resource limitations are becoming increasingly prominent due to issues such as underwater acoustic communication characteristics and node energy constraints. A resource-constrained scenario refers to situations where, due to limitations in computational power, storage space, bandwidth, power, and other aspects, there is either an insufficient supply of environmental resources or underutilization of available resources. Blockchain technology is considered an effective means to address security and privacy issues in resource-constrained environments [20]. By leveraging the decentralization, anti-data theft, traceability, and data-sharing characteristics of the blockchain, it can effectively solve security and privacy issues faced by resource-constrained devices.

We categorize the blockchain consensus security solutions into two main types: blockchain-based security solutions and blockchain consensus mechanism vulnerabilities. Table 2 contains a taxonomy of these solutions:

Li et al. [21] proposed a blockchain-based secure and energy-efficient multi-AUV collaborative operation scheme to address security concerns in underwater environments during multi-AUV target search. This scheme packages control commands and status information into blocks using blockchain's encryption algorithms, thus enhancing the

security of the solution. Yazdinejad et al. [22] introduced a blockchain-based decentralized identity authentication mechanism, which uses blockchain's distributed ledger technology to ensure secure data exchange in underwater environments.

Although the security of blockchain is widely acknowledged, there are still many security threats within its consensus mechanism. For example, attackers can secretly create a longer private chain and, by controlling over 50% of the computational power or stake, undermine the consensus mechanism in blockchain systems [23], rewriting the blockchain's original historical records and compromising the integrity of the entire blockchain. However, recent research indicates that the computational threshold required to break blockchain consensus mechanisms has significantly decreased. In [24], attackers conducted parallel Sybil attacks, requiring only 32% of the computational power to carry out a double-spending attack. Heilman et al. [25] proposed that concurrent eclipse attacks can significantly increase the likelihood of successful double-spending attacks. Therefore, the use of Proof of Work (PoW) as the consensus mechanism in blockchain systems does not fully guarantee their security. In [26], the use of group signature technology was proposed to address blockchain consensus threats. This approach involves all nodes in the cluster participating in the block signing process to defend against double-spending attacks. However, the involvement of all nodes in group signing increases the energy consumption of underwater nodes. Ramezan et al. [27] suggested increasing the number of block confirmations to defend against double-spending attacks, but this measure further exacerbates the resource consumption in underwater networks.

Table 2. Blockchain-based security solutions and their key features.

Category	Solution	Key Features	References
Blockchain-Based Security Solutions	Multi-AUV Cooperative Scheme	Enhances security using encryption algorithms in blockchain to secure multi-AUV communication	[21]
	Decentralized Identity Verification	Uses distributed ledger technology to enable secure underwater data exchange	[22]
Blockchain Consensus Mechanism Vulnerabilities	Double-Spend Attack Resistance	Double-spend attack resistance	[26]
	Increasing Block Confirmations	Increases the number of block confirmations to resist double-spend attacks	[27]

3. Network Model

The network structure of our blockchain-based collaborative node identity authentication underwater clustering model is depicted in Figure 2. This model comprises a buoy sink node positioned at the water surface, multiple underwater acoustic sensor nodes, and several AUVs [28].

In this setup, the acoustic sensor nodes are responsible for sensing and collecting environmental data. These nodes are organized using a clustering algorithm that divides them into multiple clusters. Each cluster is formed around a cluster head, which aggregates data from cluster members and transmits it upwards to the buoy sink node. The sink node, deployed at the surface, acts as an intermediary, collecting data from various cluster heads and forwarding it to a base station, which is typically located onshore or on a surface vessel. This configuration facilitates efficient data flow from the underwater environment to the surface and ultimately to end-users or processing centers.

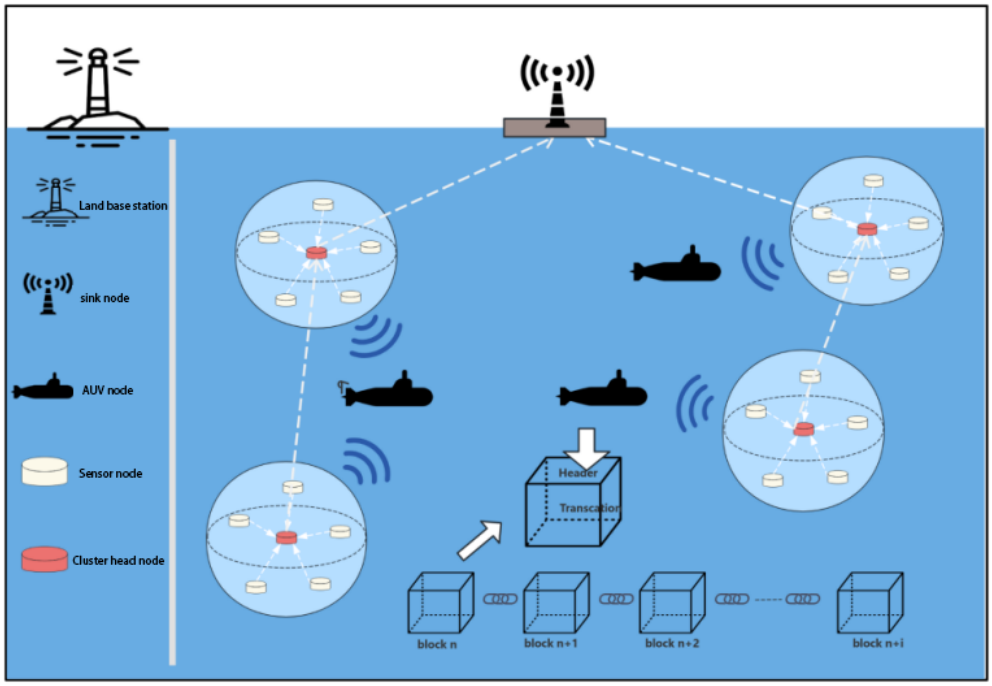


Figure 2. Network model.

Furthermore, the AUVs in this network serve dual purposes. Primarily, they function as blockchain nodes, playing a crucial role in edge computing by maintaining and authenticating the identity information of the acoustic nodes. This decentralized approach ensures robust security and authentication within the network, leveraging the inherent strengths of blockchain technology.

The abbreviations used in this paper are summarized in Table 3, facilitating an easier understanding of the terms and components within our network model.

Table 3. Abbreviations and definitions.

Parameters	Definition
MU	Normal nodes
ES	Edge computing server
RC	Registry
AS	Authorization server
GM	Group administrator
GK_i	Group i group public key
p	Proportion of cluster heads in the network
h_1	Individual learning factor
h_2	Social learning factor
w	Inertia weight
m	Population size of the particle swarm
d	Particle dimension
$T_{\max}$	Maximum number of iterations
$v_{\max}$, $v_{\min}$	Upper and lower bounds of particle velocity
r_1, r_2	Random numbers used in particle velocity calculation
x_{ij}	Position of particle i in dimension j
v_{ij}	Velocity of particle i in dimension j
p_{best}	Best position of a particle
g_{best}	Best known position among all particles

4. Blockchain-Based Collaborative Node Authentication Clustering Network Model

To address the node management and authentication issues caused by the openness of acoustic channels and the energy constraints of nodes in underwater networks, this paper proposes a blockchain-based collaborative node authentication network security model. This section first describes the clustering management of underwater acoustic nodes and the process of blockchain-assisted cluster head identity authentication. Secondly, to tackle the energy limitations of acoustic nodes, the paper improves existing clustering algorithms, optimizing them in terms of remaining energy and node density. This approach solves the "hotspot" problem caused by excessive energy consumption due to unreasonable clustering. Furthermore, the number of cluster heads is adjusted according to the current network parameters, reducing the overall energy consumption of the underwater sensor network and extending the network's lifespan.

4.1. Clustering Method for Underwater Network

4.1.1. Improved LEACH Clustering Algorithm

LEACH (Low Energy Adaptive Clustering Hierarchy) is a classic clustering scheme for energy-efficient wireless sensor networks, which is widely used in such networks. As a typical clustering algorithm, LEACH offers good scalability and adaptability. Additionally, LEACH uses a decentralized approach for cluster head selection and data transmission, reducing the reliance on centralized control, thereby enhancing the network's flexibility and robustness. Many current clustering algorithms are primarily improvements and optimizations based on the LEACH algorithm.

The classic LEACH algorithm is divided into three stages: cluster head selection, clustering, and data transmission. The basic process is as follows:

1. In the cluster head selection stage, each node generates a random constant μ $(0 < \mu < 1)$. The base station broadcasts the ratio of the previous round's cluster heads among all nodes. Afterwards, nodes compute their own election function value $T(n)$. Based on the comparison of these values, nodes decide whether to become cluster heads. If a normal node's computed value is smaller than the threshold $T(n)$, the node is selected as the cluster head for that round; otherwise, the node becomes a cluster member. The threshold formula for $T(n)$ is calculated as

$$T(n) = \begin{cases} \dfrac{p}{1 - p(rmod(\frac{1}{p}))}, n \in G \\ 0, n \notin G \end{cases} \quad (1)$$

   In Equation (1), p is the proportion of cluster heads in the entire WSN; r is the current iteration cycle; and G is the set of nodes that have not been selected as cluster heads in the last $1/p$ rounds;
2. Nodes successfully elected as cluster heads will broadcast their information, including their ID and threshold function. Non-cluster-head nodes will select the most suitable cluster to join based on the signal strength or position information from the received messages and then send a join request to the selected cluster head. The cluster head node determines whether to accept the node into the cluster. Ultimately, the network will form several clusters. Once all non-cluster-head nodes join a cluster head, the clustering process is complete;
3. In the LEACH protocol's data transmission stage, cluster members send the collected information to the cluster head using the time slots allocated by the cluster head. The cluster head aggregates the collected data and transmits the aggregated data

to the surface base station. This stage continues until the network enters the next cluster head election process, where a new cluster structure is formed to begin the next round of clustering and data transmission. In the blockchain-collaborative node authentication clustering network model, the LEACH algorithm's clustering process is shown in Figure 3.

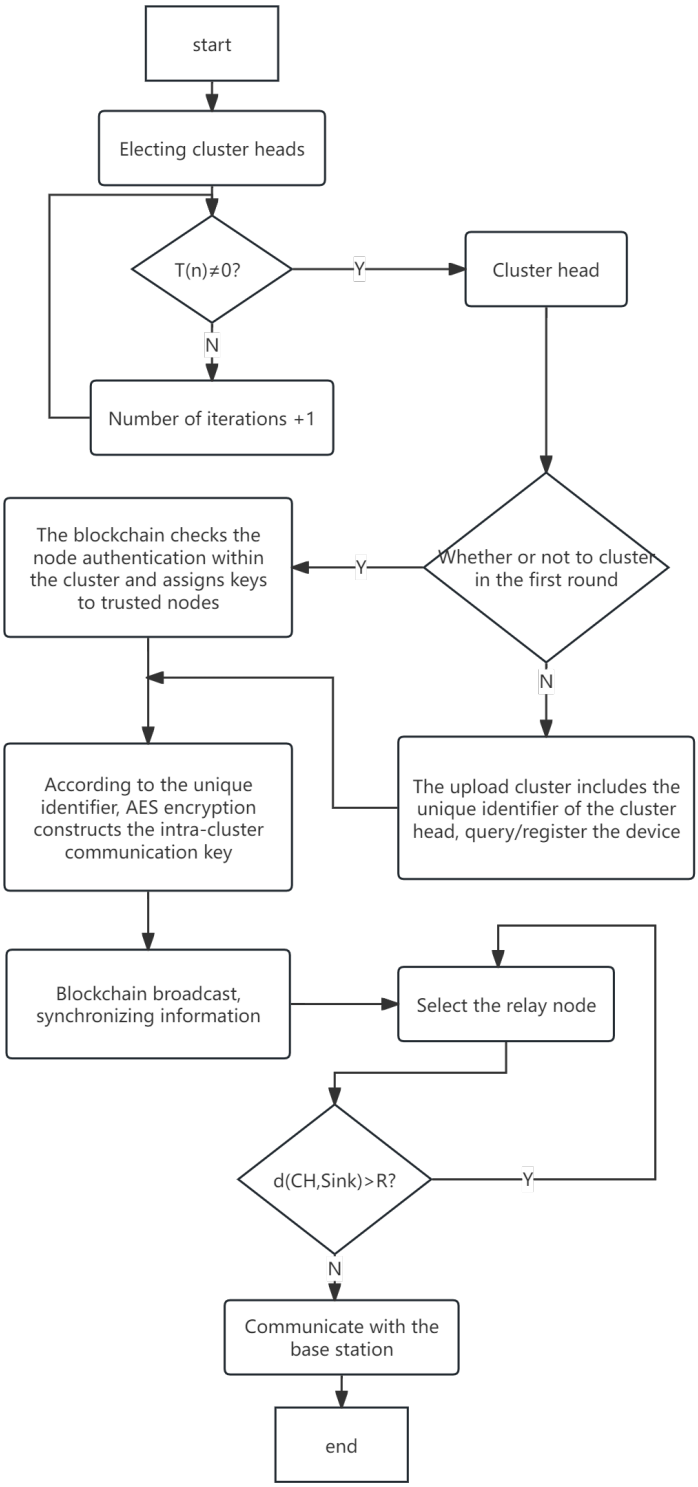


Figure 3. Flowchart of LEACH clustering algorithm in the presence of blockchain nodes.

In [29], the authors proposed an improved LEACH algorithm, which enhances the clustering process by incorporating the remaining energy factor and the node density

factor. By introducing the remaining energy factor, the algorithm considers the remaining energy of nodes, making the selection of cluster heads more reasonable. Additionally, the introduction of the node density factor ensures that nodes with fewer neighboring nodes are less suitable for becoming cluster heads. The improved clustering threshold formula is

$$T(n) = \begin{cases} \frac{p}{1-p(r \cdot \mathrm{mod}(\frac{1}{p}))}(\lambda_1 E_{re} + \lambda_2 \rho), & n \in G \\ 0, & n \notin G \end{cases} \quad (2)$$

In Equation (2), E_{re} represents the remaining energy factor, which is the ratio of the current node's remaining energy to the total energy of all nodes, and ρ is the node density factor. Nodes with fewer neighbors are less suitable for becoming cluster heads.

4.1.2. Chaos-Particle Swarm Optimization-Based LEACH Clustering Algorithm

In underwater sensor network clustering algorithms, the cluster head is a critical node responsible for collecting data from the cluster members and transmitting it to the base station [30]. The number of cluster heads (i.e., the proportion p) directly impacts the network's energy consumption and data transmission efficiency [31,32]. This paper proposes the CPSO-LEACH protocol, which optimizes the number of cluster heads (i.e., the proportion p of cluster heads in the entire network) in the LEACH-IMP protocol. The enhanced LEACH algorithm further adjusts the number of cluster heads based on the current network parameters, taking into account both the remaining energy and the node density. This adjustment aims to minimize energy consumption and prolong the network's lifespan, ensuring that the underwater clustering network does not increase the load on cluster members due to there being too few cluster heads nor does it increase the overall network energy consumption due to an excessive number of cluster heads.

The CPSO-LEACH algorithm refines the particle iteration formula. After each iteration, it computes the fitness, evaluates the clustering performance, and dynamically adjusts the number of cluster heads. The clustering strategy with the highest fitness up to the current iteration is recorded, and the evaluation continues until the iterations are completed. Ultimately, the algorithm selects the clustering strategy with the highest fitness obtained during the iterative process. This method employs the chaotic particle swarm optimization algorithm in each iteration to optimize the number of cluster heads in the LEACH algorithm. As a result, the underwater clustering network avoids both excessive load on cluster nodes due to too few cluster heads and increased overall energy consumption due to too many cluster heads. Algorithm 1 outlines the steps of the clustering of the proposed method.

1. Initialize the relevant parameters of the CPSO-LEACH algorithm, including the individual learning factor h_1, the social learning factor h_2, the inertia weight w, the population size m of the particle swarm, the particle dimension d, the maximum number of iterations $T_{\max}$, and the upper and lower bounds of particle velocity $v_{\max}$ and $v_{\min}$. The updated equations for particle velocity and position are given by

$$\begin{aligned} v_{ij}(n+1) &= w v_{ij}(n) + h_1 r_1 (p_{\mathrm{best}ij}(n) - x_{ij}(n)) + h_2 r_2 (g_{\mathrm{best}ij}(n) - x_{ij}(n)) \\ x_{ij}(n+1) &= x_{ij}(n) + v_{ij}(n+1) \end{aligned} \quad (3)$$

In Equation (3), the velocity of particle i in the j-th dimension at the $n+1$-th iteration, denoted as $v_{\mathrm{ij}}(n+1)$, is calculated by considering multiple factors: $v_{\mathrm{ij}}(n)$ is the velocity of the particle in the previous iteration, serving as the inertia component. This is multiplied by an inertia weight w to control the macroscopic and microscopic dynamics of the particle's search behavior. The term $w v_{\mathrm{ij}}(n)$ represents the influence of the previous generation's velocity on the current velocity. The inertia weight is used

to adjust the magnitude of this influence, thereby regulating the particle's search range in the solution space, achieving a balance between global and local search capabilities. The acceleration factors h_1 and h_2 are used to adjust the learning step sizes for the particle's individual historical best position and the global best position of the group, respectively. Specifically, $h_1 r_1(p_{\text{bestij}}(n) - x_{\text{ij}}(n))$ represents the individual cognitive component, while $h_2 r_2(g_{\text{bestij}}(n) - x_{\text{ij}}(n))$ represents the social cognitive component. Here, r_1 and r_2 are random numbers between 0 and 1, which are used to introduce randomness into the particle's search process.

Algorithm 1 CPSO-LEACH Clustering Algorithm.

Initialize parameters: h_1, h_2, w, swarm size m, dimensions d, max iterations $T_{\max}$, $v_{\max}$, $v_{\min}$
Initialize particle positions and velocities using Tent mapping
for each iteration n until $T_{\max}$ **do**
  for each particle $i = 1$ to m **do**
    Calculate fitness f_i using LEACH-IMP clustering
    if f_i better than p_{best} **then**
      Update p_{best}
    end if
    Update particle velocity $v_{ij}(n+1)$:

$$v_{ij}(n+1) = w \cdot v_{ij}(n) + h_1 \cdot r_1 \cdot (p_{\text{best}ij}(n) - x_{ij}(n)) + h_2 \cdot r_2 \cdot (g_{\text{best}ij}(n) - x_{ij}(n))$$

    Update particle position $x_{ij}(n+1)$:

$$x_{ij}(n+1) = x_{ij}(n) + v_{ij}(n+1)$$

  end for
  if f_i better than g_{best} **then**
    Update g_{best}
  end if
  Evaluate global best and update if necessary
end for
Output optimal clustering strategy based on best fitness

2. The Tent map strategy is used for population initialization and velocity initialization in the chaotic particle swarm optimization (CPSO) algorithm. The CPSO algorithm then calls LEACH-IMP for clustering. Once the clustering is completed, the initial fitness value is calculated, which is followed by the comparison of individual and global extrema. The iteration process for optimization begins. The fitness function is given by

$$\mathrm{f}_1 = \frac{\sum_{\mathrm{i}}^{\mathrm{n}} \mathrm{Sc_i}}{\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{Sn_i}}. \tag{4}$$

$$\mathrm{f}_2 = \frac{\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{dn_i}}{\sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{dc_i}} \tag{5}$$

$$\text{fitness}_i = \alpha_1 \cdot f_1 + \alpha_2 \cdot f_2 \tag{6}$$

In the above formula, N represents the total number of nodes, n denotes the total number of cluster heads, S_c is the remaining energy of the cluster head nodes, and S_n is the remaining energy of general nodes. In Formula (5), f is the energy evaluation factor, which represents the ratio of the total remaining energy of non-cluster-head nodes to the total remaining energy of the network. f_1 evaluates the energy dispersion of the clustering performed by the LEACH-IMP algorithm. The higher f_1 is, the higher the remaining energy of the cluster head is considered to be.

In Formula (5), f_2 is the clustering compactness evaluation factor. d_n is the distance from a regular node to the base station, and d_c is the distance from a cluster head node to the base station. f_2 represents the ratio of the sum of the distances from all nodes to the base station to the sum of the distances from competing cluster heads to the base station. The larger the value of f_2, the denser the clustering.

Formula (6) provides the fitness calculation formula for the i-th iteration, where α_1 and α_2 are impact factors, and $\alpha_1 + \alpha_2 = 1$. From the fitness calculation formula, it can be seen that clustering with higher compactness and higher remaining energy of cluster head nodes results in a higher fitness evaluation of the clustering;

3. Based on the updated fitness function formula, the individual best and global best values are updated. The positions and velocities of the particles are updated according to Formula (3), and the particle sequence undergoes chaotic optimization. The process continues until the maximum number of iterations is reached. The global best value is output, and the optimal particle best value corresponds to the optimal p value that results in the best clustering threshold.

4.2. Blockchain-Based Edge Computing Group Signature and Authentication Mechanism

4.2.1. Node Registration and Authentication

In the underwater blockchain edge computing (UBEC) mechanism, each blockchain node acts as an edge computing (EC) server, and a group of connected underwater acoustic nodes is considered as a unit with the blockchain node serving as the group manager. In the blockchain consensus process, it is assumed that the blockchain node is the node involved in the consensus algorithm (such as block creation nodes in PoS and PoW scenarios), while the underwater acoustic nodes are lightweight nodes in the blockchain and do not participate in the consensus process.

Figure 4 illustrates the network model structure of underwater blockchain edge computing, where each lightweight node can transmit its identity authentication data to the EC server in its respective group. The blockchain node performs identity authentication for each underwater node, ensuring node security during the data collection phase. Additionally, in traditional EC or IoT structures, nodes need to be authenticated through a centralized authentication server located in the cloud before operation. However, in the UBEC environment, using a centralized authentication server for each underwater node becomes cumbersome. The proposed UBEC mechanism leverages the advantages of blockchain, allowing the EC server in each group to also function as a validator for mobile devices. This decentralized authentication process reduces the complexity of identity verification by shifting the authentication process to the network edge.

Before joining the network, a new node, labeled MU, must register with the Registration Center (RC) located on the blockchain. MU begins this process by selecting an identifier, computed using the SHA256 function, where n represents the serial number of MU's mobile device. Initially, after clustering, the cluster head sends an ID list containing the identifiers of all nodes within the cluster to the RC. Based on these identifiers, specifically ID_n, the RC generates and assigns a public key PK_n and a private key SK_n for MU, which are then securely transmitted to MU.

These keys $[PK_n, SK_n]$ are first sent to the cluster head, which distributes them to each node within the cluster. This distribution completes the registration of all nodes on the blockchain.

Following registration, MU proceeds to authenticate with the Authorization Server (AS), which is also located on the blockchain. It is assumed that AS has access to all key materials stored in the RC. The authentication process involves MU sending a message to AS that includes its identifier ID_n and the identifier ID_i^{es} of the desired edge server (ES).

AS then retrieves the public key corresponding to ID_n^{mu} from the RC and sends back an encrypted message $PK_n(r)$ to MU. MU decrypts this message using its private key SK_n to extract the random number r.

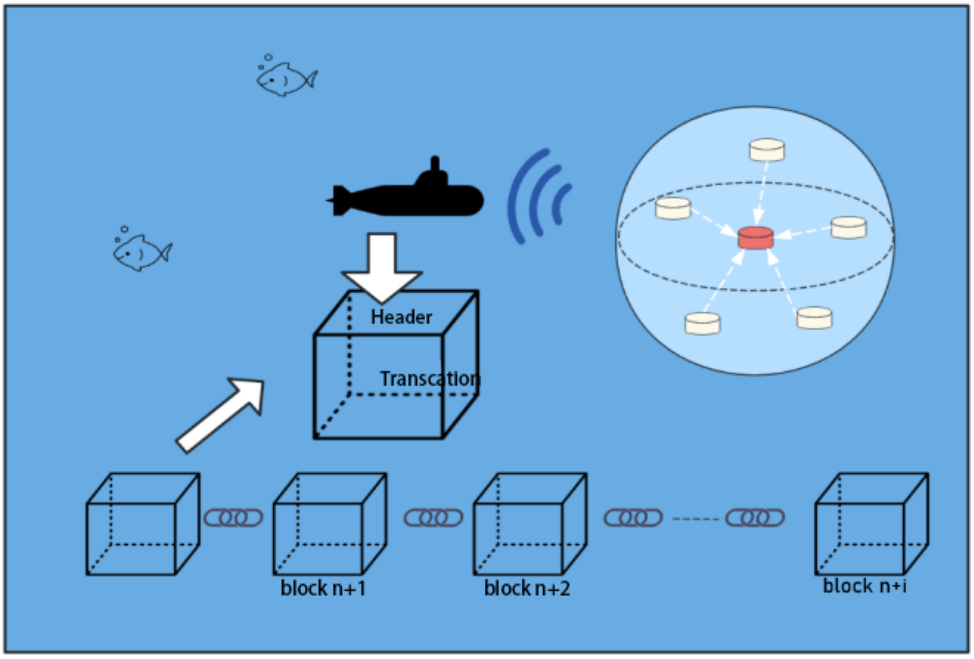


Figure 4. Network model of blockchain edge computing.

Finally, the ES decrypts the received message using its private key SK_{es} and verifies if the random number r from MU matches its own generated random number. A match confirms that MU is successfully authenticated. This authentication process is depicted in Figure 5.

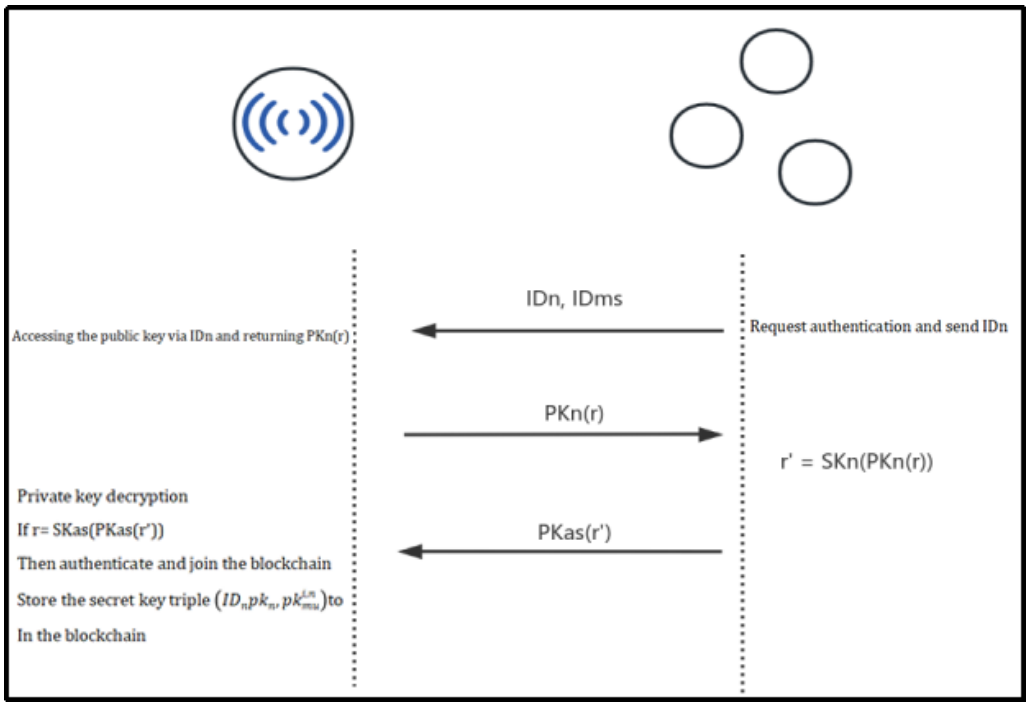


Figure 5. Node registration and verification.

The triplet is subsequently stored in the ES, while the key tuple for node n, $(ID_{\mathrm{n}}^{\mathrm{mu}}, pk_{\mathrm{n}}, pk_{\mathrm{mu}}^{\mathrm{i,n}})$, is recorded on the blockchain and broadcast across the network. This

approach allows any ES to retrieve the required key materials for node n directly from the blockchain when the MU seeks authentication with a new ES. By decentralizing the identity authentication process, this method enables the authentication data to be distributed to edge devices, facilitating efficient and decentralized identity verification for underwater acoustic nodes.

4.2.2. Group Signature and Authentication Mechanism Based on Blockchain Edge Computing

Existing group signature mechanisms typically require all nodes within a group to participate in the signing process. However, the unique challenges of underwater communication and the energy limitations of nodes make these mechanisms impractical for underwater networks. In the UBEC mechanism proposed in this paper, blockchain nodes act as group managers, while only cluster head nodes and nodes that transition between blockchain groups are designated as validation nodes. These validation nodes exclusively engage in the group signing and block validation processes. A block is deemed valid only if it secures a valid group aggregate signature from the group associated with the block creator.

Whenever a node connects to the ES, the ES stores the node's key tuple $(ID_n^{mu}, pk_n, pk_{ms})$ on the blockchain. This storage enables any new ES to retrieve the necessary key materials for the node from the blockchain when the node joins a new group and authenticates with a new ES.

Figure 6 delineates the specific steps involved in mutual authentication. A Mobile Unit (MU) from group i initiates communication by sending a message $m1 = (ID_n, TS, sk_n(ID_n, TS))$ to the ES in group $i+1$. The included timestamp TS is utilized to thwart malicious activities such as replay attacks. Upon receipt, the ES consults the blockchain for the key tuple associated with ID_n^{mu}. If the record is found, the ES uses the public key pk_n from the MU's key tuple to verify the authenticity of $sk_n(ID_n^{mu}, TS)$.

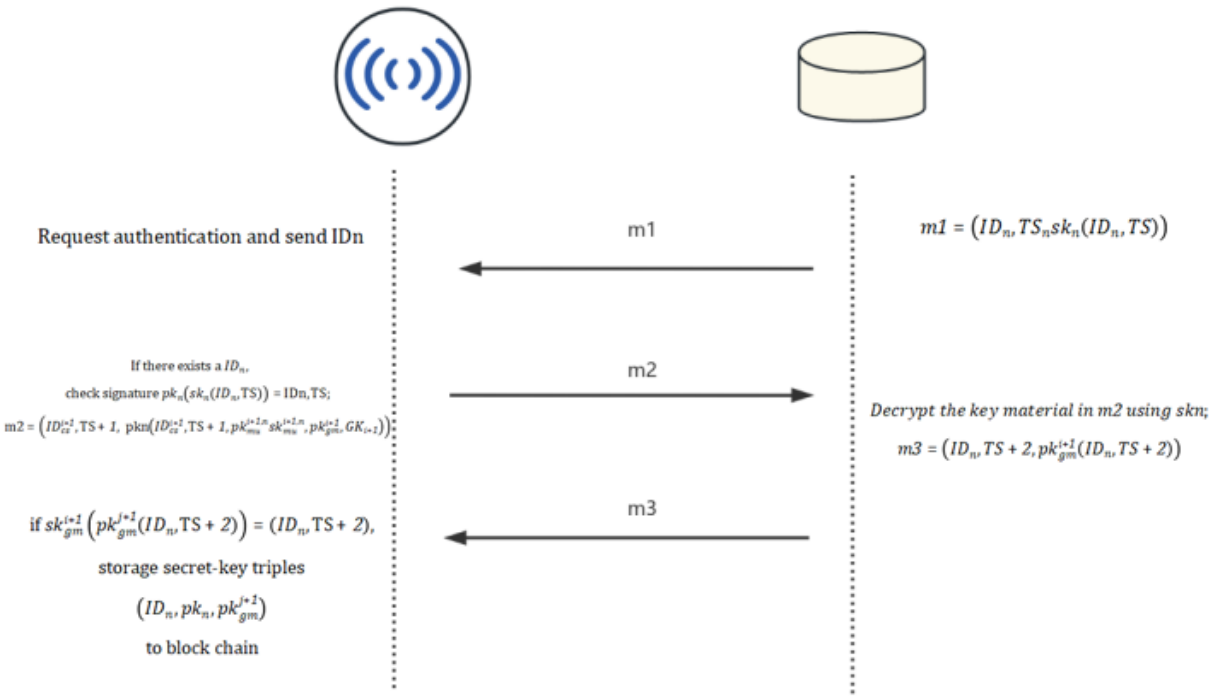


Figure 6. Verifying that the node joins the group.

If the signature from the verification node is validated, the ES sends the message $m2 = (ID_{es}^{i+1}, TS+1, pk_n(ID_{es}^{i+1}, TS+1, pk_{mu}^{i+1,n}, sk_{mu}^{i+1,n}, pk_{gm}^{i+1}, GK_{i+1}))$ to the Mobile Unit

(MU). In this instance, the MU, acting as the verification node, connects to the new ES, which functions as the group administrator (GM). The public key pk_{gm}^{i+1} denotes the new group administrator's public key. The MU decrypts the message using its private key sk_n to access the relevant key materials for the new group $i+1$.

To confirm acquisition of the new group's key materials, the MU sends a final message $m3 = (ID_{mu}^{n}, TS+2, pk_{gm}^{i+1}(ID_n, TS+2))$. Upon receipt, the ES decrypts and authenticates $m3$ using its private key sk_{gm}^{i+1}. Successful authentication grants the MU membership in the new group.

Subsequently, the ES in group $i+1$ updates and generates a new MU key tuple $(ID_n^{mu}, pk_n, pk_{mu}^{i+1,n})$, broadcasting it to the blockchain network for confirmation.

When a new verification node joins, the group public key GK_i must be updated. For group i, the ES re-aggregates the public keys of all current members using the following formula:

$$GK_i = f(pk_{mu}^{1,n}, pk_{mu}^{2,n}, pk_{mu}^{3,n}, \ldots, pk_{mu}^{i,n}) \tag{7}$$

Given the robust computational capabilities of the blockchain node, the execution time for this re-aggregation is negligible and does not impede the timely update of GK_i or affect the consensus process. After verification by other groups and appending to the longest valid chain, the updated GK_i is stored in the blockchain. Before this, the ES broadcasts the updated GK_i to other ESs for validation to ensure the authenticity of the group public key. Only after successful validation by the other ESs is GK_i recorded on the blockchain, thus ensuring that accurate group public key information is shared among the validation nodes of other groups and blockchain nodes.

4.2.3. Two-Round Block Authentication

The following describes the validation process when the blockchain node ES in group i generates a new block. When the primary node of the group i generates a new block, the block must undergo validation among the group's verification nodes. Each verification node within the group performs a validity check on the transactions stored in the block to ensure that the newly generated block contains legitimate transaction data. Once the block passes the validity check by the verification nodes, each node uses its own group private key $sk_{mu}^{i,n}$ to generate a signature $sig_{mu}^{i,n}(blkhash)$ on the block hash. This signature serves as the verification node's confirmation of the block's validity and is used in subsequent broadcasting and validation stages. To reduce the number of signatures, this paper employs the BLS aggregation signature algorithm, which aggregates the signatures generated by multiple verification nodes into a single signature. The formula is as follows:

$$\mathrm{sig_{mu}^{i}(blkhash)} = \mathrm{f(sig_{mu}^{1,n}(blkhash); \cdots ; sig_{mu}^{i,n}(blkhash))} \tag{8}$$

At the same time, the group public key GK_i of group i is updated according to Formula (8). In this way, the signatures from multiple verification nodes are aggregated into a single aggregate signature, representing the entire group's validation result for the block. This significantly reduces the number of signatures. The block can only be broadcast to other groups for a second round of validation after it has received signatures from all the members of group i, ensuring that all members of group i have confirmed and signed the block's validity before it is broadcast to the entire network.

When verification nodes from other groups receive a new block proposed by group i, they primarily assess the aggregate signature on the block hash, bypassing individual transaction verifications. This streamlined process is possible because the verification nodes within group i have already comprehensively validated the block, confirming transaction validity and generating the aggregate signature during the initial round of validation.

Consequently, this method markedly enhances the efficiency of block validation across the network.

Upon successful initial validation by group i, the block is broadcast to other groups for a second round of validation. Verification nodes in these groups need only to verify the aggregate signature on the block hash, as the detailed transaction checks have been completed by group i. This strategy significantly accelerates the validation process.

During the aggregate signature verification, nodes first fetch the latest group public key GK_i from the blockchain and use it to ascertain the validity of the aggregate signature $sig^{i}_{mu}(blkhash)$. This procedure allows nodes to focus on validating the aggregate signature without reassessing the transactions, further boosting the efficiency of block validation.

If the new block proposed by group i clears the second round of validation by other groups, it is deemed valid and appended to the longest valid chain, ensuring the integrity and continuity of the blockchain.

In summary, the 2nd round of validation effectively leverages the results of the first round, avoiding redundant verification processes. This enhances the efficiency of block validation and reduces communication overhead, contributing to the maintenance of an effective distributed ledger in resource-constrained blockchain networks. Additionally, it helps ensure the security of node identity authentication information.

5. Security and Algorithm Complexity Analysis

5.1. Security Analysis

This section evaluates potential security threats to a blockchain system operating under the longest chain rule. In traditional blockchain models without the group signature scheme, attackers could potentially create a private chain longer than the current valid chain and introduce a longer fork. According to the longest chain rule, the network would then update consensus to this longer chain, possibly leading to the loss or alteration of critical data in the original ledger.

To mitigate such risks, this paper introduces a cluster-based group signature and authentication scheme, requiring each block to undergo two rounds of validation. This dual-validation structure significantly complicates potential attacks, such as those involving forged group signatures aimed at establishing a counterfeit private chain. Attackers face two main options:

- Publishing their private chain within their group;
- Attempting to broadcast their private chain across the entire blockchain network.

For the first option, validating nodes outside the attacker's group are responsible for verifying the transactions within the latest block of the private chain. Should this external validation fail, the block will not receive the necessary signatures from group members. If the attacker tries to circumvent this initial validation phase, the block still will not secure signatures from validating nodes in other groups during the second validation round.

In both scenarios, blocks mined privately by the attacker are unlikely to obtain the required signatures from members of other groups. Even if the attacker manages to gather all signatures from their group, the aggregate signature will fail the final validation with the group's public key because it lacks essential inputs from external validating nodes.

The mathematical formulation for this analysis is as follows: Consider a group n of q members, potentially including several attacker-infiltrated nodes. Even if the majority are attackers, any honest nodes that join will act as crucial validators. For instance, if there is even one honest node in the group that does not sign the attacker's block, the absence of this single signature will significantly skew the aggregated signature, rendering it invalid.

$$\mathrm{sig}^{\mathrm{n}}_{\mathrm{mu}}(\mathrm{blkhash}) = \mathrm{f}(\mathrm{sig}^{1,\mathrm{n}}_{\mathrm{mu}}(\mathrm{blkhash}), \ldots, \mathrm{sig}^{\mathrm{q}-1,\mathrm{n}}_{\mathrm{mu}}(\mathrm{blkhash})) \quad (9)$$

Formula (9) indicates the aggregated signature missing the honest node's signature. As the group public key GK_n is updated in real time by the ES, including the honest node's public key:

$$GK_n = f(pk_{mu}^{1,n}, pk_{mu}^{2,n}, \ldots, pk_{mu}^{q,n}) \tag{10}$$

Formula (10) reflects the real-time aggregation of the group public key. The mismatch in inputs between $sig_{mu}^{n}(blkhash)$ and GK_n results in a failed signature validation during the second round of block validation, i.e., $v(G, sig_{\mu}^{n}) \neq v(GK_n, blkhash)$. Hence, any block produced by the attacker is likely to fail validation, preventing the private chain from being accepted by other network nodes.

5.2. Group Signature Mechanism Analysis

This section evaluates the effectiveness of the proposed group signature scheme in mitigating double-spending attacks within a blockchain utilizing the PoW consensus algorithm. A double-spending attack involves an attacker adding a fraudulent transaction to their private chain with the goal of extending this chain beyond the length of the main chain to replace it.

In a blockchain, where blocks are cryptographically linked, any block in the private chain that fails signature validation renders the entire private chain invalid. To succeed, the attacker must ensure that no honest Mobile Unit (MU) joins their group before the private chain surpasses the main chain. Let λ denote the probability of an honest MU joining the attacker's group within a given time frame, and let ρ represent the threshold for becoming a cluster head in the group's clustering algorithm. If an honest MU joins the group or becomes a cluster head, it will act as a validation node. Consequently, the probability of a block in the private chain being valid within a unit of time can be expressed as $(1+\rho)\cdot(1-\lambda)$.

Assume that the attacker controls a fraction α of the total computational power, while the remaining computational power of the ES is $\beta = 1-\alpha$. Let t_0 denote the average block generation time, and let z represent the number of blocks generated. The time required for the attacker to catch up with the main chain is given by $\frac{t_0}{\beta}\cdot z$. During this period, the probability that all blocks in the private chain remain valid is $(1-\lambda)^{\frac{t_0}{\beta}\cdot z}$.

This analysis adopts the probability model used to assess the success of a double-spending attack in Bitcoin, as presented in [33]. Using this model, the probability of a successful double-spending attack, P_{CG}, is calculated as

$$P_{CG} = 1 - \sum_{j=0}^{z-1}\left(\beta^z\alpha^j - \alpha^z\beta^j\right)\left(\frac{j+z-1}{j}\right) \tag{11}$$

Here, j represents the number of blocks generated by the attacker during their attempt to surpass the main chain. Building on this, the probability of a successful double-spending attack under the proposed group signature mechanism, P_{new}, is expressed as

$$P_{new} = (1+\rho)\cdot(1-\lambda)^{\frac{t_0}{\beta}\cdot z}P_{CG} \tag{12}$$

This formula integrates the probabilities of honest MU participation and the success of the attacker's private chain validation, highlighting the resilience of the proposed group signature scheme in reducing the likelihood of successful double-spending attacks.

5.3. Algorithm Complexity Analysis

In this section, we analyze the computational complexity of the group signature mechanism proposed in this study. To effectively evaluate the feasibility of our proposed

blockchain-based authentication model for UWSNs, it is crucial to analyze the computational, space, and time complexities involved.

The computational complexity of the model stems from several processes including cryptographic operations for node registration and authentication, cluster head selection, data aggregation, and block authentication. These processes involve complexities ranging from $O(n^2)$ to $O(n^3)$, depending on the cryptographic methods used, such as RSA or ECC. The cluster head selection and data aggregation processes primarily depend on the number of nodes involved, leading to complexities of $O(n)$ and $O(k)$, respectively. Block authentication, involving group signatures, adds a complexity of $O(m)$, where m is the number of nodes participating in the signature verification.

Space and time complexities are also significant considerations. Each node must store its cryptographic keys and possibly parts of the blockchain, leading to a space complexity of $O(n)$. Time complexity not only involves the computational aspects but also the latency caused by the slow underwater communication channels. Optimizing these complexities is essential for ensuring that the model is both energy-efficient and capable of operating effectively within the harsh conditions of underwater environments. This optimization is crucial for the practical deployment of secure and efficient UWSNs using blockchain technology.

6. Simulation Results

This section consists of two parts: the evaluation of the clustering algorithm and group signature authentication.

6.1. Clustering Algorithm Evaluation

To evaluate the performance of the proposed improved clustering algorithm CPOS-LEACH, CPOS-LEACH is compared with two other optimization strategies of the LEACH algorithm and the original LEACH clustering algorithm. This section provides a comparative analysis of the two optimization algorithms and the classic LEACH clustering algorithm. The experiment focuses on evaluating the performance of CPSO-LEACH in terms of network lifetime extension and energy consumption reduction in order to comprehensively assess its feasibility.

The simulation is conducted using MATLAB 2018b with the following parameters: 100 sensor nodes are randomly deployed in a 100 m × 100 m two-dimensional water area with an initial energy of 5 kJ for each node [34]. The sink node is placed at the center of the water surface. The size of all data packets is set to 500 bits. During the data transmission, the influence factors $\alpha 1$ and $\alpha 2$ for the CPSO-LEACH algorithm are set to 0.6 and 0.4, respectively. The simulation parameters are shown in Table 4.

Table 4. Experimental simulation parameter table.

Parameters	Definition
Node deployment range	100 m × 100 m
Number of nodes	100
Initial energy	5 kJ
Base station location	(50, 50)
Packet size	500 bit
($\alpha 1$, $\alpha 2$)	(0.6, 0.4)
Population size	3
Speed upper and lower limits	(5, −5)
Individual learning rate	1.49445
Group learning rate	1.49445
Number of evolutions	20
Chaos coefficient	0.5

This study examines the performance of LEACH, LEACH-E, LEACH-IMP, and CPSO-LEACH algorithms. By comparing three key performance parameters—remaining energy in the network, number of alive nodes, and network throughput—the experiment provides insights into how different algorithms affect the efficiency of underwater sensor networks from multiple perspectives.

The remaining energy in the network serves as an intuitive indicator of the energy consumption of different clustering algorithms. Once the total network energy approaches depletion, it signifies that the network's functionality will be severely impacted, and the entire network will be on the verge of collapse and becoming unusable. Figure 7 presents a comparison of the remaining energy in the network for different algorithms within the same period in the comparative experiment. Figure 8 shows the evaluation of the computational cost of different algorithms at the same time in the comparison experiment.

The results show that the LEACH algorithm depletes its energy by round 1208 with energy consumption significantly higher than the other reference algorithms. In the LEACH-E algorithm, the clustering process takes into account the remaining energy of nodes and the global average energy, prioritizing nodes with higher remaining energy as cluster heads. This results in lower energy consumption than the LEACH protocol. However, the LEACH-E protocol does not consider the impact of node distribution on energy consumption, and the network energy is depleted by round 1404.

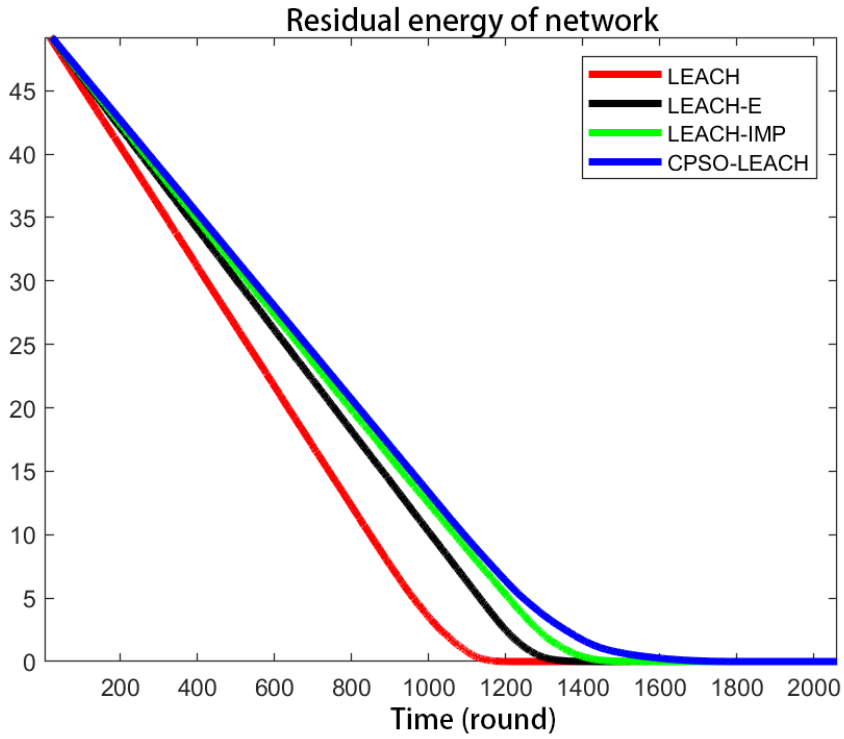


Figure 7. Residual energy of the network.

LEACH-IMP, an improved version of the LEACH algorithm, enhances the clustering process by introducing a remaining energy factor and a density factor. The modified clustering threshold formula enables nodes with higher remaining energy and more neighboring nodes to become cluster heads more easily. Compared to the LEACH-E protocol, LEACH-IMP takes node density into account, making the clustering more uniform and reducing energy consumption. It ends at round 1594.

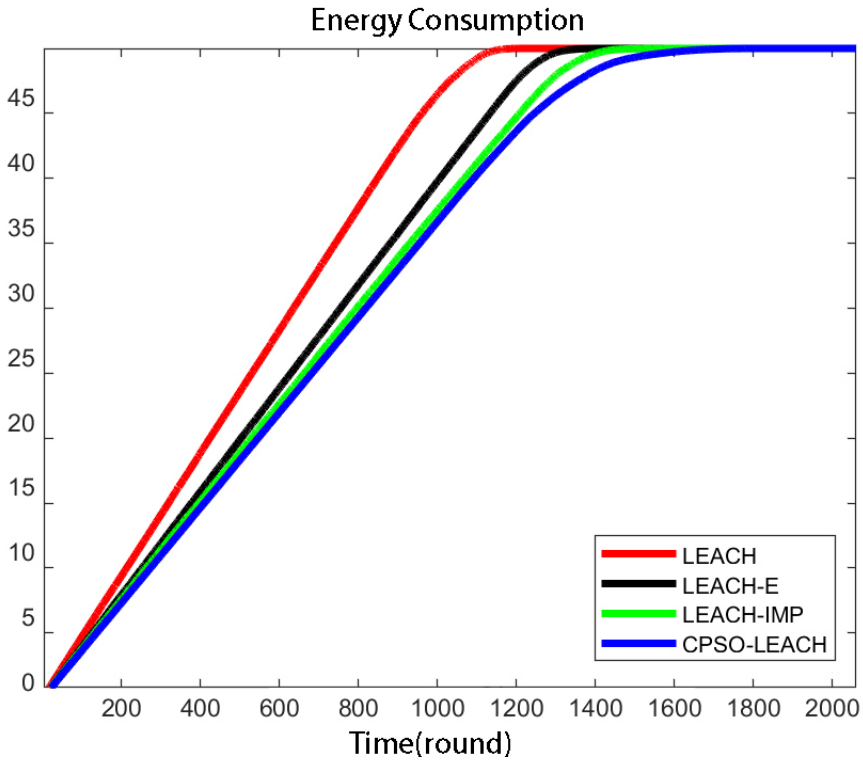


Figure 8. Energy consumption.

The CPSO-LEACH algorithm, through an iterative process, continuously updates particle positions and velocities. It adjusts particle positions based on individual and global historical optimal solutions. This feedback mechanism allows the algorithm to adapt the number of cluster heads according to current network parameters during the iteration process, optimizing energy consumption and extending the lifespan of the underwater network. The CPSO-LEACH algorithm depletes the network energy by round 1787. The simulation results are shown below.

Figure 9 shows the number of surviving nodes in the network over the same period for different algorithms. The number of surviving nodes can intuitively reflect the operational duration and efficiency of the nodes. The simulation results indicate that the number of surviving nodes in the CPSO-LEACH algorithm intersects with that of the LEACH-IMP algorithm at round 1355. After this point, the number of surviving nodes in CPSO-LEACH consistently outperforms the other three algorithms until the network energy is depleted. This is because the CPSO-LEACH algorithm utilizes a chaotic particle swarm optimization (CPSO) approach to optimize the LEACH algorithm, thus acquiring global search capability. In the later stages of the experiment, this algorithm ensures that more nodes remain operational, extending the network lifetime.

The PSO-LEACH algorithm, based on LEACH-IMP, employs chaotic particle swarm optimization, which adjusts the cluster head threshold parameters while preserving the clustering strategy that maximizes fitness. Compared to LEACH-IMP, the PSO-LEACH algorithm extends the network lifetime by 12.11%. The simulation results are shown in Figure 9.

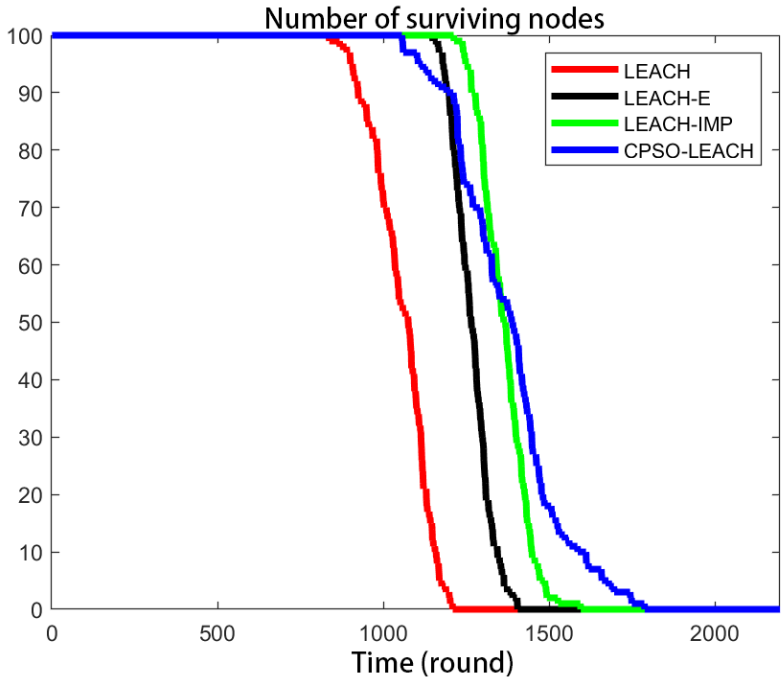


Figure 9. Number of surviving nodes.

In the network throughput comparison experiment shown in Figure 10, this section compares the performance of the LEACH, LEACH-E, LEACH-IMP, and CPSO-LEACH algorithms (with chaotic particle swarm optimization) in terms of the data received by the base station within the same period. The results show that when the number of data packets received by the base station stabilizes, the CPSO-LEACH algorithm results in the base station collecting more data packets than the other algorithms. This demonstrates the effectiveness of the CPSO-LEACH algorithm in improving energy efficiency and optimizing network data collection performance.

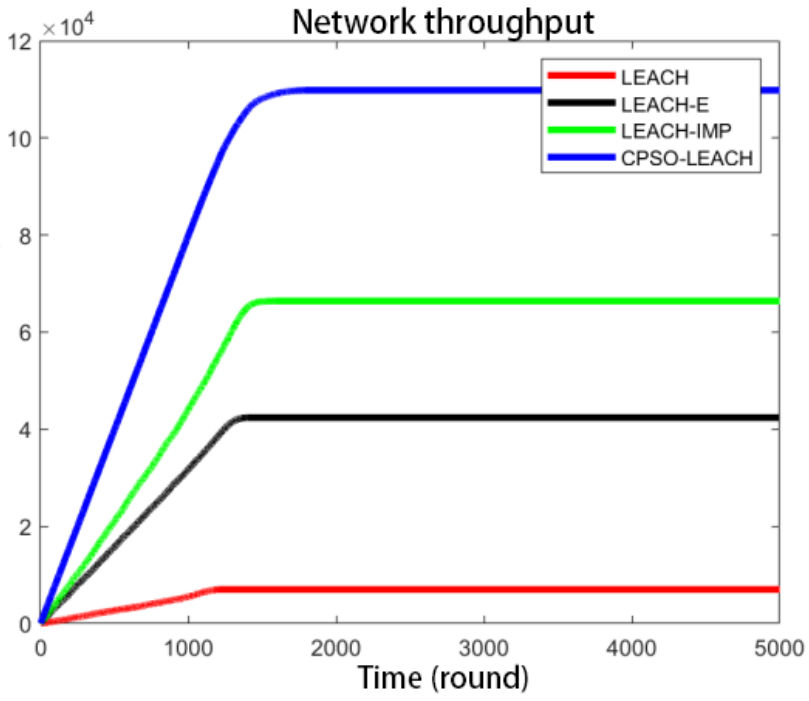


Figure 10. Network throughput.

In conclusion, the experimental results indicate that compared to the traditional improved LEACH algorithms, the CPSO-LEACH algorithm reduces network energy consumption and extends the network lifetime. It shows greater advantages in resource-constrained underwater networks with the network lifetime extended by 12.11% compared to the LEACH-IMP protocol. These results confirm that the CPSO-LEACH algorithm is well suited for data collection tasks in underwater networks.

6.2. Analysis of Resisting Double-Spend Attack

To evaluate the performance of the proposed improved clustering algorithm CPOS-LEACH, a comparative analysis was conducted between CPOS-LEACH and two other optimization strategies for LEACH as well as the original LEACH clustering algorithm. This section focuses on comparing two different optimization algorithms and the classic LEACH clustering algorithm with an emphasis on evaluating the performance of CPSO-LEACH in terms of extending network lifetime and reducing energy consumption, thereby providing a comprehensive assessment of its feasibility.

To better compare the results of the double-spending attack success probability calculated in [33] for the Bitcoin system, this study uses the real data currently adopted in the Bitcoin system, with $t_0 = 10$ min, and sets $\lambda = 0.05$ to compute P_{new}. In the experiment, the attacker possesses varying computational power. The results are compared with those obtained using Formula (11) for $\alpha = 0.2, 0.3$, and 0.4 in three different scenarios to investigate whether the proposed UNBEC mechanism can resist double-spending attacks as the attacker's computational power gradually increases. In each case, the success probability of the double-spending attack changes with z, where z ranges from 0 to 6. The reason for selecting this range is that the secure block confirmation count in the Bitcoin system is at least 6.

Since the increase in the number of block confirmations affects the transaction throughput in the blockchain, the security of the UBEC mechanism can be verified by exploring whether it is possible to reduce the number of block confirmations required to ensure transaction security. This is of significant importance for improving transaction throughput in blockchain systems, as fewer block confirmations imply faster transaction confirmation times and reduced network resource consumption.

Figure 11 shows the effectiveness of the group signature in resisting the DB attack when the attacker's computational power is 0.2. When the attacker's computational power is 0.2, the group signature method proposed in this study can achieve an attack probability equivalent to that of the traditional method without group signatures, which requires six blocks, when the number of blocks is four. This is because in the proposed group signature mechanism, once an honest node joins the attacker's group, it must participate in the new group's block signature as a verification node. Since the group's aggregated signature is in real time, rejecting the honest node's block validation would prevent the honest node from signing the new block, causing the block to be invalid. Therefore, even in the attacker's group with only a small number of honest nodes, the group signature mechanism can still effectively resist the DB attack.

As the attacker's computational power increases, the blockchain system using the group signature mechanism shows a significantly stronger ability to resist DB attacks compared to the blockchain system without the group signature mechanism. As shown in Figures 12 and 13, when the attacker's computational power is 0.3, the group signature method proposed in this study can achieve an attack probability equivalent to that of the traditional method with six blocks when the number of blocks is three. When the attacker's computational power is 0.4, the group signature method proposed in this study can achieve

an attack probability equivalent to that of the traditional method with six blocks when the number of blocks is two.

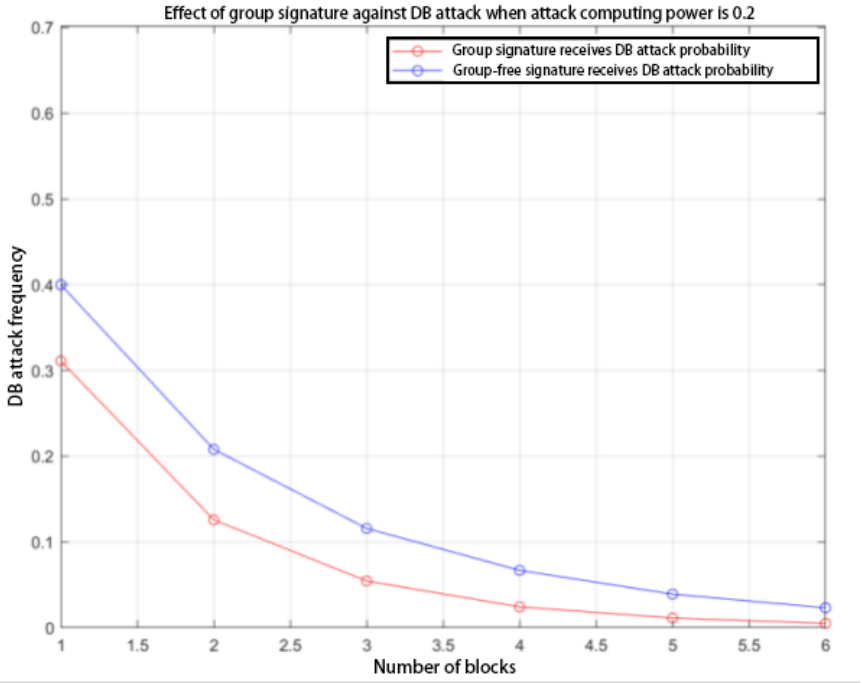


Figure 11. Effect of group signature against DB attack when attack computing power is 0.2.

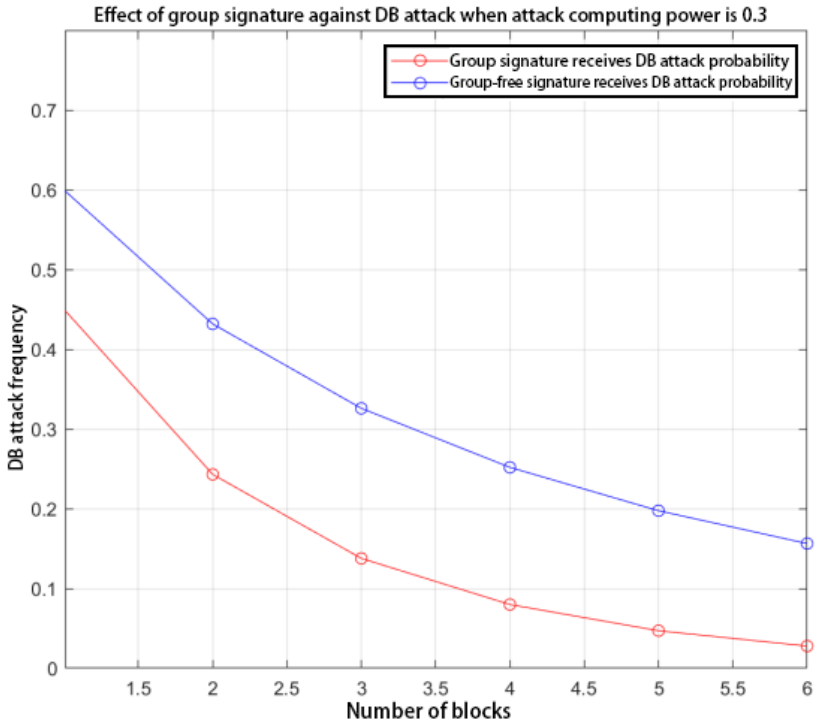


Figure 12. Effect of group signature against DB attack when attack computing power is 0.3.

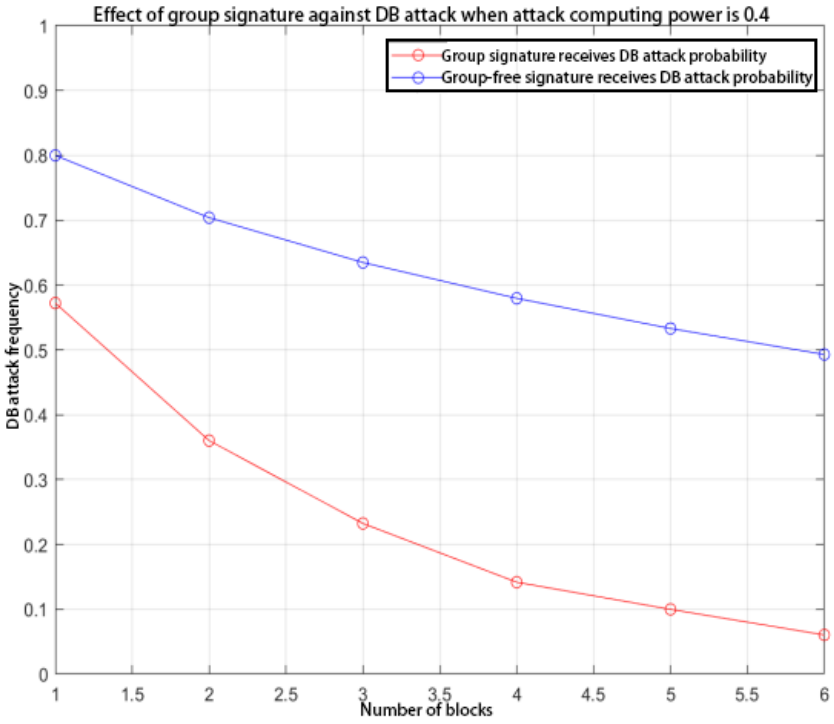


Figure 13. Effect of group signature against DB attack when attack computing power is 0.4.

Table 5 presents the effectiveness of the proposed group signature scheme in reducing the probability of DB attacks compared to traditional methods. By examining the table, specific comparison data can be obtained. When the attacker's computational power is 0.2 and the number of blocks is six, the probability of a double-spending attack in the proposed group signature method is reduced by 78.11% compared to the traditional method. When the attacker's computational power is 0.3 and 0.4, and the number of blocks is six, the probability of DB attacks is reduced by 82.23% and 87.73%, respectively.

Table 5. Group signature scheme reduces the probability of DB attack.

Number of Outgoing Blocks	Attack Hashrate = 0.2	Attack Hashrate = 0.3	Attack Hashrate = 0.4
1	22.21%	24.97%	28.48%
2	39.55%	43.70%	48.84%
3	52.94%	57.74%	63.40%
4	63.53%	68.29%	75.53%
5	70.92%	76.21%	81.26%
6	78.11%	82.23%	87.73%

From the above analysis, it can be concluded that the group signature algorithm can effectively defend against DB attacks. By reducing the number of confirmation blocks, it not only increases the efficiency of the blockchain system but also effectively resists double-spending attacks even when the attacker's computational power is relatively high, thereby enhancing the security of the blockchain.

6.3. Discussion

6.3.1. Computational Overhead of the Proposed Model

The CPSO-LEACH algorithm enhances the traditional LEACH protocol by integrating CPSO to dynamically optimize cluster head selection and network energy management. This integration significantly increases computational overhead due to the iterative nature

of CPSO, which continuously adjusts the clustering strategy based on real-time node energy levels and spatial configurations. Unlike LEACH, which uses a simple probabilistic model for cluster head selection, and LEACH-E and LEACH-IMP, which introduce static improvements for energy efficiency, CPSO-LEACH's approach is more computationally demanding. It requires extensive processing to evaluate and adjust potential solutions iteratively, aligning cluster head positions to optimize energy use and load balancing across the network.

Despite this increased computational load, CPSO-LEACH offers substantial improvements in network longevity and energy efficiency, outperforming earlier models in simulations. Its ability to dynamically respond to changes in network conditions—adapting to node failures or energy depletion—ensures more reliable and efficient operation. This makes CPSO-LEACH particularly valuable in critical underwater applications where the balance between computational resources and operational efficiency is crucial. While CPSO-LEACH incurs higher initial computational costs, the benefits in terms of extended network life and improved energy management justify the investment in more capable sensor node processors.

6.3.2. Deployment Scenarios and Operational Challenges

To effectively deploy our proposed model in UWSNs, it is essential to consider both hardware and software aspects. The hardware setup requires robust sensor nodes with advanced processing capabilities and AUVs that serve as mobile blockchain nodes, enhancing network resilience and coverage. Software requisites include a specialized operating system designed for real-time, reliable data processing and a blockchain framework that is optimized for low energy consumption and efficient performance in the demanding underwater environment. Operational challenges vary significantly depending on the deployment context: shallow waters often contend with signal interference from surface activities, while deep-sea deployments must handle extreme pressures and temperatures, requiring durable and maintenance-friendly hardware. Additionally, seismic activity in some regions necessitates adaptive network configurations to ensure stability and continuous operation. Our deployment strategy leverages adaptive clustering algorithms and machine learning to dynamically respond to environmental shifts and maintain network integrity, demonstrating the model's robust adaptability for a wide range of underwater applications.

7. Conclusions

This paper presents a novel application of blockchain technology to enhance security and node management in Underwater Wireless Sensor Networks (UWSNs). By introducing a blockchain-based collaborative node authentication model, this research addresses the dual challenges of open communication channels and limited node energy, optimizing network efficiency and longevity through an improved clustering algorithm. Additionally, it establishes a group signature and authentication mechanism tailored for blockchain edge computing, enhancing protection against consensus attacks. The effectiveness of these innovations is demonstrated through comprehensive security analyses, showcasing significant advances in the security and operational efficiency of UWSNs. Future work may explore extending these models to accommodate varying environmental conditions and assess scalability, further advancing blockchain's potential in underwater communication.

Author Contributions: Conceptualization, Y.C.; Methodology, Y.C. and R.Z.; Software, Y.C.; Validation, Y.C.; Formal analysis, Y.C. and R.Z.; Investigation, Z.L.; Resources, Z.L.; Data curation, Z.L.; Writing—original draft, Z.L.; Writing—review & editing, Z.L. and R.Z.; Visualization, R.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Hainan Provincial Natural Science Foundation of China (Grant No. 623RC449); in part by the National Natural Science Foundation of China under Grant 62362026; in part by the Key Project of Hainan Province under Grant ZDYF2023GXJS158; in part by the specific research fund of The Innovation Platform for Academicians of Hainan Province under Grant YSPTZX202314.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Xing, G.; Chen, Y.; Hou, R.; Dong, M.; Zeng, D.; Luo, J. Game-theory-based clustering scheme for energy balancing in underwater acoustic sensor networks. *IEEE Internet Things J.* **2021**, *8*, 9005–9013. [CrossRef]
2. Zhu, R.; Boukerche, A.; Long, L.; Yang, Q. Design Guidelines on Trust Management for Underwater Wireless Sensor Networks. *IEEE Commun. Surv. Tutor.* **2024**, *26*, 2547–2576. [CrossRef]
3. Busacca, F.; Galluccio, L.; Palazzo, S.; Panebianco, A.; Raftopoulos, R. Adaptive Modulation in Underwater Acoustic Networks (AMUSE): A Multi-Armed Bandit Approach. In Proceedings of the ICC 2024—IEEE International Conference on Communications, Denver, CO, USA, 9–13 June 2024.
4. Hao, Z.; Li, W.; Zhang, Q. Efficient clustering data collection in AUV-aided underwater sensor network. In Proceedings of OCEANS 2023-MTS/IEEE US Gulf Coast, Biloxi, MS, USA, 25–28 September 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6.
5. Xu, Y.; Kong, X.; Cai, Z. Cross-validation strategy for performance evaluation of machine learning algorithms in underwater acoustic target recognition. *Ocean Eng.* **2024**, *299*, 117236. [CrossRef]
6. Zhu, R.; Boukerche, A.; Huang, X.; Yang, Q. GHL-SAR: Secure and Adaptive Routing Based on GMM-HMM-LSTM for UASNs. In Proceedings of the IEEE Global Communications Conference (GLOBECOM 2023), Kuala Lumpur, Malaysia, 4–8 December 2023; pp. 4540–4545.
7. Aman, W.; Al-Kuwari, S.; Muzzammil, M.; Rahman, M.M.U.; Kumar, A. Security of Underwater and Air–Water Wireless Communication: State-of-the-Art, Challenges, and Outlook. *Ad Hoc Netw.* **2023**, *142*, 103114. [CrossRef]
8. He, M.; Liang, W.; Chen, G.; Chen, Q. Topology of underwater mobile wireless sensor networks. *Control Decis.* **2013**, *28*, 1761–1770.
9. Mertens, J.S.; Panebianco, A.; Surudhi, A.; Prabagarane, N.; Galluccio, L. Network Intelligence vs. Jamming in Underwater Networks: How Learning Can Cope with Misbehavior. *Front. Commun. Netw.* **2023**, *4*, 1179626. [CrossRef]
10. Yang, H.H.; Li, J.H.; Shen, S.; Xu, G.H. A deep convolutional neural network inspired by auditory perception for underwater acoustic target recognition. *Sensors* **2019**, *19*, 1104. [CrossRef] [PubMed]
11. Zhu, R.; Boukerche, A.; Yang, Q. An Efficient Secure and Adaptive Routing Protocol Based on GMM-HMM-LSTM for Internet of Underwater Things. *IEEE Internet Things J.* **2024**, *11*, 16491–16504. [CrossRef]
12. He, S.; Li, Q.; Khishe, M.; Salih Mohammed, A.; Mohammadi, H.; Mohammadi, M. The optimization of nodes clustering and multi-hop routing protocol using hierarchical chimp optimization for sustainable energy efficient underwater wireless sensor networks. *Wirel. Netw.* **2024**, *30*, 233–252. [CrossRef]
13. Benelhouri, A.; Idrissi-Saba, H.; Antari, J. An evolutionary routing protocol for load balancing and QoS enhancement in IoT-enabled heterogeneous WSNs. *Simul. Model. Pract. Theory* **2023**, *124*, 102729. [CrossRef]
14. Subramani, N.; Mohan, P.; Alotaibi, Y.; Alghamdi, S.; Khalaf, O.I. An efficient metaheuristic-based clustering with routing protocol for underwater wireless sensor networks. *Sensors* **2022**, *22*, 415. [CrossRef]
15. Domingo, M.C.; Prior, R. A distributed clustering scheme for underwater wireless sensor networks. In Proceedings of the 2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, Athens, Greece, 3–7 September 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 1–5.
16. Sun, H.; Pan, D.; Wang, D.; Meng, Z. LPLL-LEACH: A study of a low-power and low-delay routing protocol based on LEACH. *Ad Hoc Netw.* **2023**, *140*, 103064. [CrossRef]
17. Tian, W.; Zhao, Y.; Hou, R.; Dong, M.; Ota, K.; Zeng, D. A centralized control-based clustering scheme for energy efficiency in underwater acoustic sensor networks. *IEEE Trans. Green Commun. Netw.* **2023**, *7*, 668–679. [CrossRef]
18. Abdulzahra, A.M.K.; Al-Qurabat, A.K.M.; Abdulzahra, S.A. Optimizing energy consumption in WSN-based IoT using unequal clustering and sleep scheduling methods. *Internet Things* **2023**, *22*, 100765. [CrossRef]
19. Shah, S.M.; Hussain, T.; Shah, B.; Ali, F.; Zaman, K.; Kwak, K.S. CEER: Cooperative energy-efficient routing mechanism for underwater wireless sensor networks using clusters. *Comput. Syst. Sci. Eng.* **2023**, *45*, 2587–2602. [CrossRef]

20. Misra, S.; Mukherjee, A.; Roy, A.; Saurabh, N.; Rahulamathavan, Y.; Rajarajan, M. Blockchain at the edge: Performance of resource-constrained IoT networks. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *32*, 174–183. [CrossRef]
21. Li, X.; Guo, Y.; Yan, L.; Xu, X. Energy-aware blockchain for multiple autonomous underwater vehicles cooperative operation. In Proceedings of the 2021 40th Chinese Control Conference (CCC), Shanghai, China, 26–28 July 2021; pp. 3005–3010.
22. Yazdinejad, A.; Parizi, R.M.; Srivastava, G.; Dehghantanha, A.; Choo, K.K.R. Energy efficient decentralized authentication in internet of underwater things using blockchain. In Proceedings of the 2019 IEEE Globecom Workshops (GC Wkshps), Waikoloa, HI, USA, 9–13 December 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
23. Zheng, Z.; Xie, S.; Dai, H.; Chen, X.; Wang, H. An overview of blockchain technology: Architecture, consensus, and future trends. In Proceedings of the 2017 IEEE International Congress on Big Data (BigData Congress), Honolulu, HI, USA, 25–30 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 557–564.
24. Iqbal, M.; Matulevičius, R. Exploring sybil and double-spending risks in blockchain systems. *IEEE Access* **2021**, *9*, 76153–76177. [CrossRef]
25. Heilman, E.; Kendler, A.; Zohar, A.; Goldberg, S. Eclipse attacks on Bitcoin's peer-to-peer network. In Proceedings of the 24th USENIX Security Symposium, Washington, DC, USA, 12–14 August 2015; pp. 129–144.
26. Zhang, S.; Lee, J.H. A group signature and authentication scheme for blockchain-based mobile-edge computing. *IEEE Internet Things J.* **2019**, *7*, 4557–4565. [CrossRef]
27. Ramezan, G.; Leung, C.; Wang, Z.J. A strong adaptive, strategic double-spending attack on blockchains. In Proceedings of the 2018 IEEE International Conference on Internet of Things and IEEE Green Computing and Communications and IEEE Cyber, Physical and Social Computing and IEEE Smart Data, Halifax, NS, Canada, 30 July–3 August 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1219–1227.
28. Zhu, R.; Boukerche, A.; Li, D.; Yang, Q. Delay-aware and reliable medium access control protocols for UWSNs: Features, protocols, and classification. *Comput. Netw.* **2024**, *110631*. [CrossRef]
29. Singh, S.K.; Kumar, P.; Singh, J.P. A survey on successors of LEACH protocol. *IEEE Access* **2017**, *5*, 4298–4328. [CrossRef]
30. Khan, W.; Wang, H.; Anwar, M.S.; Ayaz, M.; Ahmad, S.; Ullah, I. A multi-layer cluster-based energy efficient routing scheme for UWSNs. *IEEE Access* **2019**, *7*, 77398–77410. [CrossRef]
31. Balamurugan, A.; Janakiraman, S.; Priya, M.D.; Malar, A.C.J. Hybrid marine predators optimization and improved particle swarm optimization-based optimal cluster routing in wireless sensor networks (WSNs). *China Commun.* **2022**, *19*, 219–247. [CrossRef]
32. Yao, Y.; Xie, D.; Li, Y.; Wang, C.; Li, Y. Routing protocol for wireless sensor networks based on Archimedes optimization algorithm. *IEEE Sens. J.* **2022**, *22*, 15561–15573. [CrossRef]
33. Grunspan, C.; Perez-Marco, R. Double spend races. *Int. J. Theor. Appl. Financ.* **2018**, *5*, 105–111. [CrossRef]
34. Zhu, R.; Liu, L.; Li, P.; Chen, N.; Feng, L.; Yang, Q. DC-MAC: A Delay-aware and Collision-free MAC Protocol Based on Game Theory for Underwater Wireless Sensor Networks. *IEEE Sens. J.* **2024**, *24*, 6930–6941. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.