

## Article

# A Lattice-Based Underwater Aggregate Signcryption Scheme

Zhehui Zhang and Ming Xu \* 

The College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China;  
202330310120@stu.shmtu.edu.cn

\* Correspondence: mingxu@shmtu.edu.cn

## Abstract

For underwater acoustic sensor networks (UWSNs), we propose a lattice-based aggregate signcryption scheme (UASC) to achieve many to one secure and efficient communication in UWSNs. Firstly, UASC constructs the pseudo identity of the node, performs lattice Gaussian sampling on the pseudo identity to generate a partial key, and then combines the underwater environmental noise collected by the node to generate a complete private key, to avoid the key generation center (KGC) leakage problem. Furthermore, UASC reuses the mathematical structures of encryption and signature on the lattice, designing a signcryption framework to achieve identity authentication and data confidentiality protection. By using the relay capability of the surface sink node, UASC combines with the ring learning with errors (RLWE) encryption and the rejection sampling to perform batch signature verification on the aggregate ciphertext set, which improves the overall response efficiency of the system. In addition, to resist tracking attacks, a dynamic update mechanism of pseudo identity and key is designed in UASC. Security analysis and experimental results show that UASC reduces overhead while satisfying relevant security requirements.

**Keywords:** lattice; signcryption; RLWE; underwater

## 1. Introduction

As an important natural space for human survival, the ocean covers more than 70% of the world's area and contains a large number of untapped resources. The exploration of the ocean is of great significance to the high quality development of coastal countries. Underwater wireless sensor networks (UWSNs) are an extension of land wireless sensor networks, which can cooperatively sense and collect data. UWSNs are widely used in the fields of marine resource exploration and protection, disaster warning, underwater vehicle navigation, and military deployments [1]. However, the development of UWSNs is relatively slow. Limited by differences in physical channels, the schemes in land wireless sensor networks cannot be directly applied to UWSNs. Considering underwater working scene, UWSNs have been facing the following problems: First, UWSNs' communication mainly relies on sound, light, radio frequency and other media, among which underwater acoustic communication has a longer range, but underwater acoustic channels have obvious multipath effects and limited bandwidth resources. Secondly, underwater nodes usually rely on batteries to provide energy [2]. Battery replacement and maintenance are difficult, and it is also difficult to use light energy to obtain energy [3]. Fully open underwater acoustic channels and unattended deployment environments make them more vulnerable to malicious attacks, such as intercepting, eavesdropping, replaying, modifying, or injecting false messages into broadcast channels to destroy data integrity and correctness in the



Academic Editor: Hung-Yu Chien

Received: 12 May 2026

Revised: 9 June 2026

Accepted: 10 June 2026

Published: 18 July 2026

**Copyright:** © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

network [4]. Attacks against UWSNs will not only pose a threat to information transmission but also pose a challenge to network availability. Therefore, in order to ensure the safe and reliable operation of UWSNs, designing an effective security mechanism has received more attention.

Currently, researchers have conducted extensive security studies on UWSNs, with most focusing on secure routing protocols at the network layer, physical layer secure communication, and secure localization. In secure routing, related studies design protocols based on node characteristics to improve network efficiency. Reference [5] proposes an energy efficient trust aware routing protocol to prolong the network lifetime. Reference [6] uses a particle swarm optimization based routing strategy to reduce transmission delay. At the physical layer, related studies exploit the randomness of underwater acoustic channels to enhance security. Reference [7] achieves key agreement on the statistical distribution of channel impulse responses. Reference [8] introduces a biomimetic covert communication scheme using dolphin clicks and phase rotation modulation for high rate covert transmission. In addition, several studies have investigated issues such as secure localization [9], access control [10], and system cooperation [11] in UWSNs. These schemes provide important support for the security and stability of UWSNs. However, most schemes focus on overall coordination and reliability, while data confidentiality and source authentication remain limited. Relying solely on secure routing or physical layer security protocols is insufficient to achieve point to point secure data transmission in UWSNs. Thus, many researchers have explored cryptographic mechanisms in UWSNs. Reference [12] designs a key agreement using hash and XOR operations. Reference [13] proposes an identity based signature with bilinear pairings, avoiding device re-authentication. Reference [14] presents a framework combining elliptic curve cryptography to verify the identity and integrity. Reference [15] develops an elliptic curve based mutual authentication using random numbers and timestamps for freshness. These studies show the importance of cryptography in UWSNs. Although these schemes improve the security of transmission to a certain extent, most of them independently investigate security functions. As an integrated system, UWSNs are required to simultaneously satisfy multiple security properties. Therefore, investigating how to achieve both data confidentiality and identity authentication in UWSNs with low cost is significant.

Signcryption, a combination of encryption and signature, is considered to be an effective means to solve security and privacy problems in resource-constrained environments. In recent years, many scholars have studied the application of signcryption in wireless sensor networks. Reference [16] proposed a certificateless aggregate signcryption scheme based on the CDHP problem and the elliptic curve discrete logarithm problem, which can discover illegal ciphertexts during the aggregation phase by constructing a two dimensional matrix. In [17], a zero-knowledge proof-based certificateless encryption scheme without bilinear pairing, which guarantees confidentiality, unforgeability, and public verification for secure communication. Reference [18] proposed a certificateless aggregate signcryption scheme that introduced the Chinese remainder theorem into blockchain-based signcryption, enabling batch verification after receiving multiple messages. Reference [19] proposed a certificateless aggregate signcryption scheme based on blockchain that provides identity binding and verifiable identity auditing. In [20], a revocable certificate-based broadcast signcryption scheme supporting edge computing is proposed, in which the base station sends ciphertexts to the edge server, and the edge server re-encrypts the ciphertexts for legitimate nodes according to the revocation list.

The above schemes show the exploration of communication security in wireless sensor networks from many aspects, but they are still based on traditional cryptography problems. Shor's algorithm [21] shows that they will no longer be secure in the quantum era. At

the same time, post-quantum cryptography has made rapid progress, and lattice-based cryptography has become a research hotspot. Reference [22] proposes a NTRU-based key encapsulation mechanism to achieve worst-case correctness error in a moderate modulus. Reference [23] presents an identity based aggregate signature system on lattices to improve the efficiency of the verification process. Reference [24] proposes an NTRU-based certificateless signature scheme and proves it secure in the random oracle model. Reference [25] proposed a lattice-based signcryption scheme and provided the formal proof for lattice-based signcryption. Reference [26] proposes a signcryption scheme for medical management, which combines the threshold method with signcryption to provide anonymity for internal members. Reference [27] proposes an identity based linkable ring signcryption scheme for IoT devices, which reduces the key length to adapt to lightweight applications in resource-trapped environments. Reference [28] proposes a blockchain based ring signcryption scheme for vehicular ad hoc networks to ensure timely message relay, vehicle anonymity, and linkable privacy. Reference [29] proposes a lattice-based lightweight signcryption scheme for the IoT medical environment to ensure the anonymity and unlinkability of users on the blockchain. In [30], a lattice-based certificateless online/offline signcryption scheme is proposed in which the offline phase reduces the computational burden of the online phase. Reference [31] proposed a lattice-based linkable ring signature, incorporates a binary tree structure and a lattice-based key conversion algorithm to support periodic key updates, thereby providing forward security.

The above schemes have explored signcryption schemes that can resist quantum attacks for different scenarios. However, most of the existing signcryption schemes do not take into account the scenarios of UWSNs. How to achieve a balance between post quantum security, low round interaction, lightweight implementation and underwater environmental adaptability is still a key problem to be solved in the current UWSN security research. Based on this, this paper aims to study the post quantum underwater signcryption scheme for the practical application scenarios of UWSNs, focusing on solving the problems of existing schemes in terms of anti-quantum security, protocol overhead control under resource-constrained conditions, key center leakage prevention, and transmission coordination in the sensor network. In order to ensure security such as identity authentication, message confidentiality and integrity, the computational and communication burden during the operation of the protocol is reduced as much as possible, to improve the applicability and engineering feasibility of the scheme in UWSNs. Based on previous studies [22,24], we propose a lattice-based underwater aggregate signcryption scheme for UWSNs. The following are the main contributions of this paper:

- A cryptographic key generation mechanism combining Gaussian sampling over the NTRU trapdoor and underwater channel-assisted randomness is proposed to alleviate the key escrow problem of the key generation center (KGC). Since the KGC cannot obtain the complete private key generated with the participation of underwater environmental randomness, this mechanism can alleviate the key escrow problem of the KGC and enhance the randomness of the node key generation process.
- A heterogeneous lattice-based signcryption framework for UWSNs is constructed to achieve confidentiality, authentication, and integrity in a unified signcryption process. The sender and the receiver adopt different lattice-based key structures, and message encryption and signature authentication are completed simultaneously in one logical step. This design can reduce protocol interaction and computational overhead while ensuring the required security properties.
- A lattice-based underwater aggregate signcryption scheme (UASC) is designed for scenarios where multiple nodes upload data simultaneously. The sink node is used to perform aggregate signatures on the set of valid ciphertexts. This mechanism can

reduce the repeated verification overhead and improve the response efficiency and scalability of UWSNs in this scenario.

- A dynamic pseudonym and key update mechanism is introduced to enhance anonymity and unlinkability in node communication. By switching among different valid pseudonyms and updating the corresponding keys, it can reduce the probability that an attacker links multiple communication behaviors to the same node.
- This paper provides a formal security proof of UASC in the random oracle model (ROM). Specifically, the confidentiality against chosen-ciphertext attacks is proved based on the ring learning with errors (RLWE) problem, and the existential unforgeability against chosen-message attacks is proved based on the RSIS problem. In addition, this paper analyzes the performance of UASC in terms of security properties, computational overhead, ciphertext length, and batch decryption overhead under different network scales, so as to verify its feasibility in resource-constrained UWSN environments.

The organization of this paper is as follows: Section 2 briefly introduces the lattice basis and related concepts. Section 3 firstly introduces the communication model and security objectives of UWSNs and then shows the specific algorithm of UASC. Section 4 analyzes the confidentiality, unforgeability and unlinkability of UASC. Section 5 compares UASC with other advanced schemes in terms of computational overhead and communication overhead and summarizes the main findings.

## 2. Preliminaries

**Definition 1** ((Lattice) [32]). A lattice  $\Lambda$  is a discrete set of points in Euclidean space generated by all integer linear combinations of linearly independent basis vectors. Given  $v_1, \dots, v_n$  be a set of linearly independent vectors on the  $m$ -dimensional Euclidean space  $\mathbb{R}^m$  and the coefficients satisfying  $a_i \in \mathbb{Z}$ . Lattice  $\Lambda$  is defined as follows:  $L(\Lambda) = \{\sum_{i=1}^n a_i v_i \mid a_i \in \mathbb{Z}, i = 1, \dots, n\}$ . Where  $V = \{v_1, \dots, v_n\}$  is the set of basis of the lattice  $\Lambda$ ,  $n$  is the rank of  $\Lambda$  and  $m$  is the dimension of  $\Lambda$ . Lattice structures constitute the algebraic foundation of the proposed scheme and provide quantum-resistant security.

**Definition 2** ((RSIS problem) [33]). The RSIS problem in the NTRU lattice  $\Lambda_{h,q}$  is described as follows: Given  $R_q = \mathbb{Z}_q[x]/(x^n + 1)$ , two polynomials  $f, g \in R_q$  on  $D_{\mathbb{Z},\sigma}$ , set  $h = g/f \in R_q^\times$  and  $A = (h, 1) \in R_q^{1 \times 2}$ . The goal is to find a solution  $(z_1, z_2) \in \Lambda_{h,q}$  such that  $\|(z_1, z_2)\| \leq \beta$  and  $A(z_1, z_2)^T = 0$ . The hardness of the RSIS assumption can be reduced to solving the shortest vector problem (SVP) over ideal lattices. The RSIS problem provides security support for the signature of our signcryption scheme and guarantees its unforgeability against adaptive attacks.

**Definition 3** ((RLWE problem) [33]). Given the polynomial ring  $R_q = \mathbb{Z}_q[x]/(x^n + 1)$ , where the integer  $q > 2$ ,  $n = 2^k \geq 1$  and  $k$  is an integer, let  $\chi$  be an error distribution over  $R_q$ . Sample an error  $e$  from  $\chi$ , and choose  $a \in R_q$  uniformly at random. For a secret value  $s \in R_q$ , the RLWE distribution is defined as

$$A_{s,\chi} = (a, b = as + e \bmod q) \in R_q \times R_q. \quad (1)$$

1. Search RLWE problem: For any independent instance  $(a, b = as + e \bmod q) \in R_q \times R_q$ , the goal is to recover  $s$  from this instance.
2. Decisional RLWE problem: For an independent instance  $(a, b) \in R_q \times R_q$ , the goal is to distinguish, with non-negligible probability, whether the instance  $(a, b)$  is sampled from the RLWE distribution  $A_{s,\chi}$  or from the uniform distribution  $U$  over  $R_q \times R_q$ .

The hardness of the RLWE assumption can be reduced to solving the shortest independent vector problem over ideal lattices. The RLWE problem provides the confidentiality foundation of the proposed signcryption scheme. Its algebraic structure enables efficient computation and compact parameter representation, which are desirable for resource-constrained underwater sensor nodes.

**Definition 4** (Rejection sampling). *Rejection sampling is mainly used to make the distribution of the signature output as independent of the private key information as possible. Reference [34] proposed a lattice signature with rejection sampling. Specifically, a vector  $y$  is first sampled from the distribution  $D$ . Then, compute  $c \leftarrow H(Ay, \mu)$  and  $z = Sc + y$ , and the signature  $(c, z)$  is output with probability  $\min(\frac{D_{\sigma}^m(z)}{MD_{Sc, \sigma}^m(z)}, 1)$ . Where,  $A$  denotes the public key,  $S$  denotes the private key, and  $\mu$  denotes the message. This enables the distribution of the signature  $(c, z)$  to be as independent of the secret key  $S$  as possible. Our scheme introduces the ring variant of the rejection sampling based signature mechanism in [35] to enhance the protection of the signature component against private key information leakage during the signcryption process.*

Algorithm 1: The NTRU trapdoor algorithm is a class of trapdoor generation algorithms constructed based on the NTRU lattice structure. Reference [32] is taken as an example, the specific algorithm is shown in Algorithm 1. The NTRU lattice trapdoor generation algorithm is used to efficiently generate the public and private key pairs in the scheme, so as to realize the construction of a more flexible signcryption scheme for UWSNs.

---

**Algorithm 1** NTRU trapdoor generation algorithm.

---

**Input:**  $n, q \in \mathbb{Z}, \sigma > 0$

**Output:**  $B \in \mathbb{R}^{2n \times 2n}, h \in R_q^\times$

- 1: Sample  $(g, f) \leftarrow D_{\mathbb{Z}^n, \sigma}$  such that  $f \in R_q^\times$  and  $g \in R_q^\times$ . If  $\|f\| > \sigma\sqrt{n}$  or  $\|g\| > \sigma\sqrt{n}$  or  $\langle f, g \rangle \neq R$ , restart.
  - 2: Compute  $F_1, G_1 \in R$  such that  $fG_1 - gF_1 = 1$ , compute  $F_q = qF_1$  and  $G_q = qG_1$ .
  - 3: Approximate  $(F_q, G_q)$  by a linear combination of  $(f, g), (xf, xg), \dots, (x^{n-1}f, x^{n-1}g)$ .
  - 4: Output  $(F, G)$  such that there exists  $k \in R$  with  $(F, G) = (F_q, G_q) - k(f, g)$ . If  $\|(F, G)\| > n\sigma$ , restart.
  - 5: Return  $B = \begin{bmatrix} C(f) & C(g) \\ C(F) & C(G) \end{bmatrix}$  and  $h = g/f \in R_q^\times$ .
- 

**Algorithm 2:** The Gaussian sampling algorithm is a randomized algorithm that samples from the integer domain or lattice space according to a discrete Gaussian distribution. Reference [36] is taken as an example, the specific algorithm is shown in Algorithm 2. Gaussian sampling is employed to generate statistically secure lattice vectors, thereby enhancing the security and reliability of the proposed scheme.

---

**Algorithm 2** Gaussian sampling algorithm.

---

**Input:** A basis  $B$  of an  $n$ -dimensional lattice  $\Lambda$ , standard deviation  $\sigma > 0$ , and center  $c \in \mathbb{Z}^n$

**Output:** A vector  $v$  sampled from  $D_{\Lambda, \sigma, c}$

- 1: Initialize  $v_n \leftarrow 0$  and  $c_n \leftarrow c$
  - 2: **for**  $i \leftarrow n, \dots, 1$  **do**
  - 3:   Compute  $c'_i = \langle c_i, \tilde{b}_i \rangle / \|\tilde{b}_i\|^2$  and  $\sigma'_i = \sigma / \|\tilde{b}_i\|$
  - 4:   Sample  $z_i$  from algorithm  $\text{SampleZ}(\sigma'_i, c'_i)$
  - 5:   Compute  $c_{i-1} = c_i - z_i b_i$  and  $v_{i-1} = v_i + z_i b_i$
  - 6: **end for**
  - 7: **return**  $v_0$
-

**Algorithm 3:** A classical single-threshold quantization method. After the quantization threshold  $d$  is determined, the specific method is as follows in Algorithm 3:

---

**Algorithm 3** single threshold quantization

---

**Input:** the quantization threshold  $d$ , sequence  $[h_1, h_2, \dots, h_N]$ .

**Output:**  $Q(h_i) = \begin{cases} 0, & h_i < d, \\ 1, & h_i > d, \end{cases} \quad i = 1, 2, \dots, N.$

---

This method has a faster quantization rate, but the generated quantized bits are relatively simple. Therefore, a 0–1 bit sequence with higher randomness can be obtained through privacy amplification. This algorithm quantizes the channel feature sequence in the key generation phase.

**Definition 5** ((Gaussian distribution) [37]). The  $m$ -dimension Gaussian function centered on the vector  $c \in \mathbb{R}^m$ ,  $x \in \mathbb{R}^m$  and variance parameter  $s > 0$  is defined as follows:  $\rho_{s,c}(x) = \exp(-\pi\|x - c\|^2/s^2)$ . For the lattice  $\Lambda \in \mathbb{R}^m$ , the discrete Gaussian distribution is defined as  $D_{\Lambda,s,c}(x) = \rho_{s,c}(x)/\rho_{s,c}(\Lambda)$ , where  $\rho_{s,c}(\Lambda)$  is the sum of  $\rho_{s,c}(x)$  on lattice  $\Lambda$ . The following two lemmas will be used in the proof of unforgeability to show the existence of non-unique solutions.

**Lemma 1.** For  $\sigma > 0$ ,  $m > 0$  and  $x \in \mathbb{R}^m$ , the conclusions hold:

1.  $\Pr[x \leftarrow D_\sigma^1 : |x| > 12\sigma] < 2^{-100}.$
2.  $\Pr[x \leftarrow D_\sigma^m : \|x\| > 2\sigma\sqrt{m}] < 2^{-m}.$

**Lemma 2** ([34]). For  $v \in \mathbb{Z}^m$  and  $\alpha > 0$ , the conclusions hold:

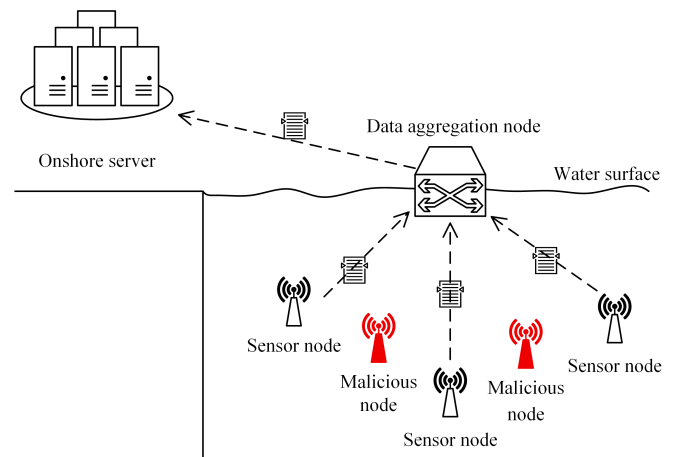
1.  $\Pr\left[x \leftarrow D_\sigma^m : \frac{D_\sigma^m(x)}{D_{\sigma,v}^m(x)} = O(1)\right] = 1 - 2^{-\omega(\log m)}, \sigma = \omega(\|v\|\sqrt{\log m}).$
2.  $\Pr\left[x \leftarrow D_\sigma^m : \frac{D_\sigma^m(x)}{D_{\sigma,v}^m(x)} < e^{12/\alpha+1/(2\alpha^2)}\right] > 1 - 2^{-100}, \sigma = \alpha\|v\|.$

**Definition 6** ((Forking Lemma) [38]). For a digital signature scheme with security parameter  $k$ , if there exists a polynomial-time adversary  $\mathcal{A}$  that can forge a valid signature  $(m, \sigma_1, h, \sigma_2)$  on a message  $m$  within the time bound  $T$  with probability  $\epsilon \geq 7Q/2^k$ , where  $Q$  is the number of queries made by  $\mathcal{A}$  to the random oracle. Then, there exists another algorithm that uses  $\mathcal{A}$  as a subroutine and can replay the execution of  $\mathcal{A}$  while changing the value of  $h$ . In this way, it can generate two valid signatures  $(m, \sigma_1, h, \sigma_2)$  and  $(m, \sigma_1, h', \sigma_2')$  for the same message  $m$ , where  $h \neq h'$ . The expected running time of this algorithm is bounded by  $T' \leq 84480TQ/\epsilon$ . The Forking Lemma is adopted to provide theoretical support for the security reduction proof of the unforgeability of the proposed scheme under adaptive chosen message attacks.

### 3. Materials and Methods

#### 3.1. System Model

The UWSNs defined in this paper are shown in Figure 1, including underwater sensor nodes (SNs), the Data Aggregation Node (DA), the Trusted Authority (TA) and the Onshore Server (OS). The sink DA floating on the water surface has more abundant computing and storage resources than SNs sinking into the water. Both of them are equipped with OFDM modems for communication. As KGC, TA is responsible for the initialization of the system, generating public parameters and partial private keys of nodes. SNs in the same area modulate the encrypted acquisition information into an OFDM signal and uploads it to the DA. The DA authenticates the aggregated message and forwards it to the OS; the OS receives messages, verifies in batches and decrypts ciphertext at the same time. During this period, there are attackers by default, and attackers can intercept transmission messages and impersonate legitimate nodes to forge messages.



**Figure 1.** UWSN acquisition and upload system model.

### 3.2. Communication Model

We assume that the nodes in UWSNs are equipped with OFDM modems for underwater acoustic communication. OFDM has strong resistance to multipath effects and high spectrum efficiency. After receiving the OFDM signal, the receiver converts the time-domain signal into frequency-domain discrete samples through the fast Fourier transform (FFT). Then, the received information is recovered through channel estimation, Doppler compensation, and other operations. The received signal can be expressed as

$$y(t) = x(t) * h(t) + n(t), \quad (2)$$

where  $x(t)$  denotes the transmitted signal,  $h(t)$  denotes the channel state information (CSI), and  $n(t)$  denotes the additive ocean noise [39]. Furthermore, the CSI of the underwater acoustic channel can be expressed as

$$h(t) = \sum_l^L a_l e^{j2t\pi\theta_l} \delta(\tau - \tau_l), \quad (3)$$

where  $L$  denotes the number of multipath components,  $a_l$  denotes the amplitude component,  $\theta_l$  denotes the Doppler frequency shift,  $\tau_l$  denotes the time delay, and  $\delta(\cdot)$  denotes the Dirac delta function. After FFT, the channel frequency response (CFR) of the  $k$ -th subcarrier in OFDM can be estimated as

$$\tilde{H}(f_k) = H(f_k) + \tilde{N}(f_k). \quad (4)$$

The underwater acoustic channel is seriously affected by multipath propagation and the Doppler effect. Meanwhile,  $h(t)$  has rapid time-varying characteristics [40] and spatial uniqueness [41], which can be used to extract features for key generation [42]. Ocean noise  $n(t)$  comes from various sources. It is usually treated as a stochastic signal and can be used as the source of a random number generator [43]. Therefore, while the OFDM modem completes the basic task of underwater acoustic communication, we can also use the underwater environmental noise collected by the modem to enhance the randomness of our scheme. By quantizing the channel estimation sequence  $[h_1, h_2, \dots, h_N]$  affected by the underwater environmental noise, a highly randomized bit sequence can be obtained and used in the generation of the public-private key pair of the node. This mechanism is used to enhance the environment-related randomness of node key generation, while the core provable security of the scheme is still guaranteed by the lattice hard problems and cryptographic hash functions.

### 3.3. Security Model

During information transmission in UWSNs, both the overall system and individual sensor nodes are confronted with various security and privacy threats. Adversaries may launch attacks from different directions. Firstly, a selected node acts as the KGC in UWSNs, while this node is also restricted by underwater resource constraints. Once the KGC node is compromised by adversaries, private keys will suffer centralized leakage. The approach that enables each node to independently generate its full private key can prevent KGC from holding complete private key data and enhance system security. Secondly, if the real identities of SNs are transmitted directly in the network, adversaries can perform identity forgery attacks or malicious tracing attacks. Adversaries can infer sensitive information by linking consecutive sessions belonging to the same node. Applying a pseudonym mechanism with periodic updates can realize the unlinkability of node identities, thus protecting user privacy. Thirdly, since UWSNs adopt open wireless communication channels, adversaries are able to conduct eavesdropping, message forgery and man-in-the-middle (MitM) attacks. Therefore, it is necessary to ensure security properties including data confidentiality and message integrity. In order to protect the secure communication of UWSNs, the security model of UASC is considered from the following perspectives:

1. **Correctness:** Whether the generation and decryption process of signcryption ciphertext is effective and whether the batch verification process after ciphertext aggregation is effective.
2. **Message confidentiality:** The adversary can directly intercept the transmitted ciphertext and public parameters but cannot obtain any information about the plaintext.
3. **Unforgeability:** The receiver will determine the legitimacy of the ciphertext source by verifying the signature on the ciphertext. The adversary can learn and forge the signature but cannot forge a valid signature.
4. **Unlinkability:** In order to protect privacy and cope with long-term tracking learning, malicious attackers cannot link different sessions to the same SN.

### 3.4. Scheme Definition

Our scheme mainly includes the following seven parts:

1. **System initialization:** In the system initialization phase, given the security parameters, the TA generates the master private key and discloses the master public key and public parameters.
2. **Registration node:** The purpose of this phase is to grant nodes the legal identity to participate in UWSNs, and any sensor node must register with the TA before communicating with other entities.
3. **Signcryption:** In this stage, the SN encrypts and transmits the collected sensitive data to the DA, and the DA is aggregated and forwarded to the OS.
4. **Unsigncryption:** When the receiver receives the ciphertext group, it performs the verification-then-decryption execution paradigm.
5. **Aggregate signcryption:** In this stage, the DA, as a transit node, collects the ciphertext sent by the SN in the area and generates an aggregate signature.
6. **Batch decryption:** The OS verifies the aggregate ciphertext in batches at the same time and then decrypts.
7. **Pseudonym and key update:** In this stage, a new pseudonym and public-private key pair are generated with the help of the TA.

### 3.5. Concrete Algorithm

#### 3.5.1. System Initialization

Let the prime modulus be  $q = \tilde{\Omega}(\beta\sqrt{n})$  with  $q > 5$ ,  $s = \Omega\left(\frac{q}{n}\sqrt{\ln(8nq)}\right)$ , and  $\sigma = 12s\kappa n$  for Gaussian sampling, where  $\kappa$  is a positive integer [24]. Define the integer polynomial ring as  $R = \mathbb{Z}[x]/(x^n + 1)$ , and denote its quotient ring modulo  $q$  by  $R_q = R/(q) = \mathbb{Z}_q[x]/(x^n + 1)$ . The elements in  $R_q$  are polynomials of degree less than  $n$  whose coefficients are taken from the finite field  $\mathbb{Z}_q$ . Let  $f = \sum_{i=0}^{n-1} f_i x^i$  and  $g = \sum_{i=0}^{n-1} g_i x^i$  be polynomials over  $R$ , and let  $f * g = fg \bmod (x^n + 1)$ . On input  $\lambda$ , TA runs Algorithm 1 to generate a polynomial  $P = g/f$  and a trapdoor basis  $B \in \mathbb{Z}_q^{2n \times 2n} = R_q^{2 \times 2}$ , where  $B = \begin{bmatrix} C(f) & C(g) \\ C(F) & C(G) \end{bmatrix}$ ,  $P \in R_q^*$ , and  $R^*$  denotes the set of invertible elements in  $R$ . Set  $P$  and  $B$  as the master public key ( $mpk$ ) and the master secret key ( $msk$ ) of the system. Select five collision-resistant hash functions.  $H_1 : \mathbb{Z}_q^n \times \{0, 1\}^* \rightarrow \{0, 1\}^l$ ,  $H_2 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^n$ ,  $H_3 : \mathbb{Z}_q^n \times \{0, 1\}^* \rightarrow \mathbb{Z}_q^n$ ,  $H_4 : \mathbb{Z}_q^n \rightarrow \{0, 1\}^l$ ,  $H_5 : \{0, 1\}^* \times \mathbb{Z}_q^n \rightarrow \mathbb{Z}_q^n$ . Output the public parameters  $pp = \{q, s, \sigma, n, P, H_1, H_2, H_3, H_4, H_5\}$ , and keep  $msk = B$ .

#### 3.5.2. Registration Node

$SN_i$  sends a registration request  $\{\text{Reg\_rep}, \text{real identity } (RID_i)\}$  to the TA. The TA checks whether  $RID_i$  has been registered. If it has not been registered, the TA randomly selects  $y \in R_q^*$  and computes  $\theta = yP + e$ , the pseudo identity  $PID_i = H_1(RID_i || \theta)$ , where the coefficients of  $e \in R_q$  are sampled from the narrow distribution  $\rho$  and  $RID_i$  denotes the real identity of  $SN_i$ . The TA runs Algorithm 2 to get  $(s_{i1}, s_{i2}) \leftarrow (t_i, 0)\text{-Gaussian\_Sampler}(B, \sigma_i, (t_i, 0))$ , where  $t_i = H_2(PID_i)$ . Then, the TA sends  $(s_{i1}, s_{i2}, PID_i)$  to  $SN_i$ .  $SN_i$  verifies whether  $\|(s_{i1}, s_{i2})\| \leq s\sqrt{2n}$  and  $s_{i1} + s_{i2}P = H_2(PID_i)$  hold. If the verification succeeds, let  $s_i = (s_{i1}, s_{i2})$ .  $SN_i$  collect the channel state sequence  $[h_1, h_2, \dots, h_N]$  affected by underwater environmental noise. Then, it obtains a bit sequence by using a quantization method similar to Algorithm 3 and derives the partial private key  $(x_{i1}, x_{i2})$  through privacy amplification with the hash function  $H : \{0, 1\}^* \rightarrow \mathbb{Z}_q^n \times \mathbb{Z}_q^n$ , sets  $x_i = (x_{i1}, x_{i2})$ , and computes  $x_{i1} + x_{i2}P = P_i$ . Finally, set  $sk_i = (s_i, x_i)$ ,  $pk_i = (P_i, H_2(PID_i))$ . Following the design principle of NTRU, OS randomly samples  $(r_1 \in R_q, r_2 \in R_q) \leftarrow \chi$ , and computes  $d = 3r_1 + 1$ ,  $h = 3r_2 d^{-1}$  [22]. Then, set  $sk_r = d$ ,  $pk_r = h$ .

#### 3.5.3. Signcryption

Randomly sample the error parameters  $e_i \leftarrow \chi$ ,  $\alpha_{i1} \leftarrow \chi$ ,  $\beta_{i1} \leftarrow \chi$ ,  $\alpha_{i2} \leftarrow \chi$ ,  $\beta_{i2} \leftarrow \chi$ . Compute  $u_i = H_3(RID_i || m_i, v_i)$ , where  $RID_i$  denotes the real identity of  $SN_i$ ,  $m_i$  denotes the message, and  $v_i$  denotes the random sequence extracted from the underwater acoustic channel. Compute  $y_i = pk_r u_i + e_i$ ,  $c_i = (RID_i || m_i) \oplus H_4(e_i)$ . Compute  $w_i = H_5\left(\begin{bmatrix} \alpha_{i1}P + \beta_{i1} \\ \alpha_{i2}P + \beta_{i2} \end{bmatrix}, c_i\right)$ ,  $\begin{bmatrix} z_{i1} \\ z_{i2} \\ z_{i3} \\ z_{i4} \end{bmatrix} = \begin{bmatrix} s_{i2} \\ s_{i1} \\ x_{i2} \\ x_{i1} \end{bmatrix} * w_i + \begin{bmatrix} \alpha_{i1} \\ \beta_{i1} \\ \alpha_{i2} \\ \beta_{i2} \end{bmatrix}$ . Let  $z_i = [z_{i1}, z_{i2}, z_{i3}, z_{i4}]^T$ . Output  $(z_i, w_i)$  with probability  $\min\left(1, \frac{D_{R,\sigma}(z_i)}{MD_{R,s,\sigma}(z_i)}\right)$ , where  $s = [s_{i2}, s_{i1}, x_{i2}, x_{i1}]^T * w_i$ . Finally, output the ciphertext  $ct_i = \{y_i, c_i, w_i, z_i\}$  and send it to the DA.

#### 3.5.4. Unsigncryption

Compute  $t_i = \begin{bmatrix} z_{i1}P + z_{i2} \\ z_{i3}P + z_{i4} \end{bmatrix} - \begin{bmatrix} H_2(PID_i) \\ P_i \end{bmatrix} * w_i$ ,  $\hat{w}_i = H_5(t_i || c_i)$ . Verify whether  $\hat{w}_i = w_i$  holds. If it holds, continue to execute the following algorithm; otherwise, return  $\perp$ .

Compute  $e_i = (y_i sk_r \bmod q) \bmod 3$ . Compute  $RID_i \| m_i = c_i \oplus H_4(e_i)$ , and accept the message  $(RID_i, m_i)$ .

### 3.5.5. Aggregate Signcryption

DA verifies  $ct_i = \{y_i, c_i, w_i, z_i\}$  from the underwater nodes. If the verification succeeds, DA randomly samples  $y_1^* \leftarrow \chi, y_2^* \leftarrow \chi$ . Compute  $c^* = \{c_i\}_{i=1}^n, y^* = \{y_i\}_{i=1}^n, w^* = H_5(y_1^* P + y_2^* \| c^*), \begin{bmatrix} z_1^* \\ z_2^* \end{bmatrix} = \begin{bmatrix} s_{DA,2}^* \\ s_{DA,1}^* \end{bmatrix} * w^* + \begin{bmatrix} y_1^* \\ y_2^* \end{bmatrix}$ . Generate the aggregate ciphertext  $ct^* = \{w^*, z_1^*, z_2^*, c^*, y^*\}$  and forward it to OS.

### 3.5.6. Batch Decryption

Compute  $t' = z_1^* P + z_2^* - H_2(PID_{DA}) w^*, w' = H_5(t' \| c^*)$ . Verify whether  $w' = w^*$  holds. If it holds, the OS runs the Unsigncrypt algorithm to decrypt and obtain the identity set of SNs  $\{RID_i\}_{i=1}^n$  and the message set  $\{m_i\}_{i=1}^n$ . Otherwise, return  $\perp$ .

### 3.5.7. Pseudonym and Key Update

$SN_i$  sends an update request  $\{\text{Upd\_rep}, RID_i\}$  to the TA. The TA queries the tuple corresponding  $\{PID_i, RID_i, \theta_i\}$  to  $RID_i$  in its memory. Then, the TA randomly selects  $y^{new} \in R_q^*, e^{new} \leftarrow \rho$ , computes  $\theta^{new} = y^{new} P + e^{new}$ , and generates a new pseudo identity  $PID_i^{new} = H_1(PID_i \| \theta^{new})$ . The TA runs  $(s_{i1}^{new}, s_{i2}^{new}) \leftarrow \text{Gaussian\_Sampler}(B, \sigma_i, (t_i^{new}, 0))$ , where  $t_i^{new} = H_3(PID_i^{new})$ . Then, the TA sends  $(s_{i1}^{new}, s_{i2}^{new}, PID_i^{new})$  to  $SN_i$ .  $SN_i$  verifies whether  $\|(s_{i1}^{new}, s_{i2}^{new})\| \leq s\sqrt{2n}$  and  $s_{i1}^{new} + s_{i2}^{new} P = H_3(PID_i^{new})$  hold. If the verification succeeds,  $SN_i$  collects the underwater noise sequences to get  $x_{i1}^{new}, x_{i2}^{new}$  and sets  $s_i^{new} = (s_{i1}^{new}, s_{i2}^{new}), x_i^{new} = (x_{i1}^{new}, x_{i2}^{new})$ . Finally,  $SN_i$  sets the new private–public key pair as  $sk_i = (s_i^{new}, x_i^{new})$  and  $pk_i = (p_i^{new} = x_{i1}^{new} + x_{i2}^{new} P, PID_i^{new})$ .

### 3.6. Correctness Analysis

For a single valid ciphertext  $ct_i = \{y_i, c_i, w_i, z_i\}$ , the verification process is given as follows:

$$\begin{aligned}
 t_i &= \begin{bmatrix} z_{i1}P + z_{i2} \\ z_{i3}P + z_{i4} \end{bmatrix} - \begin{bmatrix} H_2(PID_i) \\ P_i \end{bmatrix} w_i \\
 &= \begin{bmatrix} z_{i1}P + z_{i2} - H_2(PID_i)w_i \\ z_{i3}P + z_{i4} - P_i w_i \end{bmatrix} \\
 &= \begin{bmatrix} z_{i1}P + z_{i2} - (s_{i1} + s_{i2}P)w_i \\ z_{i3}P + z_{i4} - (x_{i1} + x_{i2}P)w_i \end{bmatrix} \\
 &= \begin{bmatrix} (s_{i2}w_i + \alpha_{i1})P + (s_{i1}w_i + \beta_{i1}) - (s_{i1} + s_{i2}P)w_i \\ (x_{i2}w_i + \alpha_{i2})P + (x_{i1}w_i + \beta_{i2}) - (x_{i1} + x_{i2}P)w_i \end{bmatrix} \\
 &= \begin{bmatrix} (s_{i1} + s_{i2}P)w_i + \alpha_{i1}P + \beta_{i1} - (s_{i1} + s_{i2}P)w_i \\ (x_{i1} + x_{i2}P)w_i + \alpha_{i2}P + \beta_{i2} - (x_{i1} + x_{i2}P)w_i \end{bmatrix} \\
 &= \begin{bmatrix} \alpha_{i1}P + \beta_{i1} \\ \alpha_{i2}P + \beta_{i2} \end{bmatrix}.
 \end{aligned} \tag{5}$$

When  $\hat{w}_i = w_i$ , the signature verification succeeds, and the decryption is executed as follows:

$$\begin{aligned}
 e_i &= ((y_i sk_r \bmod q) \bmod 3) \\
 &= ((3r_2 u_i + e_i d \bmod q) \bmod 3) \\
 &= ((3r_2 u_i + 3e_i r_1 + e_i \bmod q) \bmod 3) \\
 &= e_i.
 \end{aligned} \tag{6}$$

Therefore, the plaintext  $(RID_i, m_i)$  can be obtained by computing

$$RID_i || m_i = c_i \oplus H_4(e_i). \quad (7)$$

For an aggregate ciphertext set  $ct^* = \{w^*, z_1^*, z_2^*, ct^*\}$ , the verification process is given as follows:

$$\begin{aligned} t' &= z_1^*P + z_2^* - H_2(PID_{DA})w^* \\ &= (s_{DA,2}^*w^* + y_1^*)P + s_{DA,1}^*w^* + y_2^* - (s_{DA,1}^* + s_{DA,2}^*P)w^* \\ &= y_1^*P + y_2^*. \end{aligned} \quad (8)$$

If the aggregate signature verification  $w' = w^*$  succeeds, the same decryption as that used for a single ciphertext is subsequently executed.

## 4. Security Proof

Theorem 1 proves the confidentiality of UASC based on the RLWE problem, and Theorem 2 proves the unforgeability of UASC based on the RSIS problem. These two theorems characterize two core security goals of the signcryption scheme from the perspective of reduction proofs. By linking the structure of UASC with the lattice hard problems, UASC can achieve post-quantum resistance.

### 4.1. Message Confidentiality

**Theorem 1.** *In the ROM, if there exists a probabilistic polynomial-time adversary  $\mathcal{A}$  that can break the Indistinguishability under Chosen-Ciphertext Attack (IND-CCA) security of the proposed scheme with a non-negligible probability  $\xi$ , then there exists a challenger  $\mathcal{C}$  that can solve the RLWE problem with advantage  $\xi^* \geq (1 - q_{H_4}q_d/2^n)\xi$ , where  $q_{H_4}$  denotes the maximum number of  $H_4$  hash queries and  $q_d$  denotes the maximum number of decryption oracle queries.*

**Proof of Theorem 1.** To solve a random RLWE instance  $(a, u = as + e)$ ,  $\mathcal{C}$  uses it as a subroutine in its interaction with  $\mathcal{A}$  and records the corresponding query results in lists during the interaction.

Initialization.  $\mathcal{C}$  executes the initialization algorithm, sets  $pk_r^* = a$  and sends  $pk_r^*, \Sigma = (H_1, H_2, H_3, H_4, H_5, n, q, h)$  to  $\mathcal{A}$ . Meanwhile,  $\mathcal{C}$  maintains a set of lists  $\{L_1, L_2, L_3, L_4, L_5\}$  to simulate the random oracles queried, and these lists are initially empty. Then,  $\mathcal{A}$  adaptively performs the following queries.

Phase 1. When  $\mathcal{A}$  performs adaptive queries,  $\mathcal{C}$  responds as follows.

$H_1$  hash query.  $\mathcal{C}$  maintains a list  $L_1 = \{RID_i, \theta, h_1\}$  to simulate the random oracle  $H_1$ . When  $\mathcal{A}$  queries  $(RID_i, \theta)$ ,  $\mathcal{C}$  first searches whether the tuple exists in the list. If it exists,  $\mathcal{C}$  returns  $h_1$ ; otherwise,  $\mathcal{C}$  randomly selects  $h_1 \in \{0, 1\}^n$ , returns it and updates it in the list  $L_1$ .

$H_2$  hash query.  $\mathcal{C}$  maintains a list  $L_2 = \{PID_i, s_{i1}, s_{i2}, h_2\}$  to simulate the random oracle  $H_2$ . When  $\mathcal{A}$  queries  $PID_i$ ,  $\mathcal{C}$  first searches whether the tuple exists in the list. If it exists,  $\mathcal{C}$  returns  $h_2$ ; otherwise,  $\mathcal{C}$  randomly selects  $s_{i1} \in R_q$  and  $s_{i2} \in R_q$ , computes  $h_2 = s_{i1} + s_{i2}P$ , returns  $h_2$  and updates  $\{PID_i, s_{i1}, s_{i2}, h_2\}$  in the list  $L_2$ .

$H_3$  hash query.  $\mathcal{C}$  maintains a list  $L_3 = \{RID_i, m_i, v_i, h_3\}$  to simulate the random oracle  $H_3$ . When  $\mathcal{A}$  queries  $(RID_i, m_i, v_i)$ ,  $\mathcal{C}$  first searches whether the tuple exists in the list. If it exists,  $\mathcal{C}$  returns  $h_3$ ; otherwise,  $\mathcal{C}$  randomly selects  $h_3 \in \mathbb{Z}_q^n$ , returns it and updates it in the list  $L_3$ .

$H_4$  hash query.  $\mathcal{C}$  maintains a list  $L_4 = \{e_i, h_4\}$  to simulate the random oracle  $H_4$ . When the adversary queries  $e_i$ ,  $\mathcal{C}$  first searches whether the tuple exists in the list. If it exists,

$\mathcal{C}$  returns  $h_4$ ; otherwise,  $\mathcal{C}$  randomly selects  $h_4 \in \{0, 1\}^n$ , returns it and updates it in the list  $L_4$ .

$H_5$  hash query.  $\mathcal{C}$  maintains a list  $L_5 = \left\{ \begin{bmatrix} \alpha_{i1}P + \beta_{i1} \\ \alpha_{i2}P + \beta_{i2} \end{bmatrix}, c_i, h_5 \right\}$  to simulate the random oracle  $H_5$ . When  $\mathcal{A}$  makes a query,  $\mathcal{C}$  first searches whether  $\left( \begin{bmatrix} \alpha_{i1}P + \beta_{i1} \\ \alpha_{i2}P + \beta_{i2} \end{bmatrix}, c_i, h_5 \right)$  exists in the list. If it exists,  $\mathcal{C}$  returns  $h_5$ ; otherwise,  $\mathcal{C}$  randomly selects  $h_5 \in \mathbb{Z}_q^n$ , returns it and updates it in the list  $L_5$ .

Decryption query. When receiving a decryption query for ciphertext  $(y_i, c_i, w_i, z_i)$  under key pair  $(pk_s, sk_s)$ ,  $\mathcal{C}$  first searches the lists  $L_2$  and  $L_4$  to find a tuple  $(u_i, e_i)$  such that  $y_i = pk_r^* u_i + e_i$  holds, in order to return a valid plaintext  $(RID_i, m_i)$ . Then,  $\mathcal{C}$  queries the list  $L_3$  to determine whether the tuple  $(RID_i, m_i, v_i, u_i)$  exists. If it exists,  $\mathcal{C}$  checks whether  $c_i = (RID_i || m_i) \oplus H_4(e_i)$  holds. If it does not hold,  $\mathcal{C}$  returns  $\perp$ ; otherwise, it returns  $(RID_i, m_i)$ .

Challenge phase. After Phase 1 ends,  $\mathcal{A}$  presents two different messages  $(RID_0 || m_0)$  and  $(RID_1 || m_1)$  of the same length, and designates the sender's public-private key pair  $(pk_s^*, sk_s^*)$  for a query. The challenger  $\mathcal{C}$  randomly chooses a bit  $b \in \{0, 1\}$ , randomly selects  $w_i \in R_q$  and  $c^* \in \{0, 1\}^l$ , sets  $(z_i^*)^T \in (R_q, R_q, R_q, R_q)$ , and lets  $y^* = u$ . Then,  $\mathcal{C}$  returns the challenge ciphertext  $ct^* = (y^*, c^*, w_i^*, z_i^*)$  to  $\mathcal{A}$ .

Phase 2.  $\mathcal{A}$  continues to make polynomially many adaptive queries as in Phase 1, but it is not allowed to perform an unsigncryption query on ciphertext  $ct^*$  with the designated key pair  $(pk_s^*, sk_s^*)$  to access the associated plaintext.

Guess. Based on the above interactions,  $\mathcal{A}$  outputs a result  $b'$ . If  $b' = b$ ,  $\mathcal{A}$  emerges as the victor in the game. Note that, after receiving  $ct^*$ ,  $\mathcal{A}$  cannot confirm the  $ct^*$ 's validity unless it has queried the hash values  $H_4(e)$  and  $s = H_3(RID_i || m_i, v_i)$ . Meanwhile, the solution to the RLWE instance is exactly embedded into  $L_3$  and  $L_4$ . Therefore,  $\mathcal{C}$  can search for each tuple  $(u_i, e_i)$  in the lists  $L_3$  and  $L_4$ , verify whether  $y^* = pk_r^* u_i + e_i$  holds. If the equation holds,  $\mathcal{C}$  then outputs  $(u_i, e_i)$  as a solution to the RLWE instance.

Consider  $B$  denotes an event, where  $\mathcal{A}$  has queried  $s = H_3(RID_i || m_i, v_i)$  and  $H_4(e)$ . We can obtain  $\Pr[b = b'] = \Pr[b = b' | \neg B] \Pr[\neg B] + \Pr[b = b' | B] \Pr[B] \leq \frac{1}{2} + \frac{1}{2} \Pr[B]$ , and thus  $\xi = |2 \Pr[b = b'] - 1| \leq \Pr[B]$ . During  $\mathcal{C}$ 's simulation, a valid ciphertext may be rejected in the unsigncryption query. Since there is one tuple  $(RID_j, m_j, v_j, h_3 = u_i)$  in the oracle  $H_3$  that can generate a valid ciphertext for  $(e_i, h_4)$ , the probability of rejecting a valid ciphertext is not more than  $q_{H_4} q_d / 2^n$ . As a result, there exists a probability  $\xi^* \geq (1 - q_{H_4} q_d / 2^n) \xi$  that  $\mathcal{C}$  can solve the RLWE problem. However, the RLWE problem is considered to be hard to solve. Therefore, the adversary  $\mathcal{A}$  cannot break the IND-CCA security of the proposed scheme.  $\square$

#### 4.2. Unforgeability

**Theorem 2.** *If the RSIS problem is hard over lattice rings, then the proposed scheme is Existential Unforgeability under Chosen-Message Attack (EUF-CMA) security.*

**Proof of Theorem 2.** Suppose that there exists a polynomial-time adversary  $\mathcal{A}$  that can forge a valid ciphertext of the proposed scheme with a non-negligible probability. Then, there exists a polynomial-time challenger  $\mathcal{C}$  that can solve the RSIS problem by using its interaction with  $\mathcal{A}$  as a subroutine. To achieve this goal, the following query-response operations are executed between  $\mathcal{A}$  and  $\mathcal{C}$ .

$H_1$  oracle query.  $\mathcal{C}$  maintains a list  $L_1 = \{RID_i, \theta, h_1\}$  to simulate the random oracle  $H_1$ . When  $\mathcal{A}$  queries  $(RID_i, \theta)$ ,  $\mathcal{C}$  first checks whether the corresponding tuple exists in

the list. If it exists,  $\mathcal{C}$  returns  $h_1$ ; otherwise,  $\mathcal{C}$  randomly selects  $h_1 \in \{0, 1\}^n$ , returns it, and updates it in the list  $L_1$ .

$H_2$  oracle query.  $\mathcal{C}$  maintains a list  $L_2 = \{PID_i, s_{i1}, s_{i2}, h_2\}$  to simulate the random oracle  $H_2$ . When  $\mathcal{A}$  queries  $PID_i$ ,  $\mathcal{C}$  first checks whether the corresponding tuple exists in the list. If it exists,  $\mathcal{C}$  returns  $h_2$ ; otherwise,  $\mathcal{C}$  randomly selects  $s_{i1} \in R_q$  and  $s_{i2} \in R_q$ , computes  $s_{i1} + s_{i2}P = h_2$ , returns  $h_2$ , and updates  $\{PID_i, s_{i1}, s_{i2}, h_2\}$  in the list  $L_2$ .

$H_3$  oracle query.  $\mathcal{C}$  maintains a list  $L_3 = \{RID_i, m_i, v_i, h_3\}$  to simulate the random oracle  $H_3$ . When  $\mathcal{A}$  queries  $(RID_i, m_i, v_i)$ ,  $\mathcal{C}$  first checks whether the corresponding tuple exists in the list. If it exists,  $\mathcal{C}$  returns  $h_3$ ; otherwise,  $\mathcal{C}$  randomly selects  $h_3 \in Z_q^n$ , returns it, and updates it in the list  $L_3$ .

$H_4$  oracle query.  $\mathcal{C}$  maintains a list  $L_4 = \{e_i, h_4\}$  to simulate the random oracle  $H_4$ . When  $\mathcal{A}$  queries  $e_i$ ,  $\mathcal{C}$  first checks whether the corresponding tuple exists in the list. If it exists,  $\mathcal{C}$  returns  $h_4$ ; otherwise,  $\mathcal{C}$  randomly selects  $h_4 \in \{0, 1\}^n$ , returns it, and updates it in the list  $L_4$ .

$H_5$  oracle query.  $\mathcal{C}$  maintains a list  $L_5 = \left\{ \begin{bmatrix} \alpha_{i1}P + \beta_{i1} \\ \alpha_{i2}P + \beta_{i2} \end{bmatrix}, c_i, h_5 \right\}$  to simulate the random oracle  $H_5$ . When  $\mathcal{A}$  makes a query,  $\mathcal{C}$  first searches whether  $\left( \begin{bmatrix} \alpha_{i1}P + \beta_{i1} \\ \alpha_{i2}P + \beta_{i2} \end{bmatrix}, c_i, h_5 \right)$  exists in the list. If it exists,  $\mathcal{C}$  returns  $h_5$ ; otherwise,  $\mathcal{C}$  randomly selects  $h_5 \in Z_q^n$ , returns it, and updates it in the list  $L_5$ .

Public key query.  $\mathcal{C}$  maintains a list  $L_k = \{PID_i, s_i, x_i, P_i, pk_i\}$ . When  $\mathcal{A}$  queries  $PID_i$ ,  $\mathcal{C}$  first checks whether the tuple corresponding to  $PID_i$  already exists. If it exists,  $\mathcal{C}$  returns  $pk_i$ ; otherwise,  $\mathcal{C}$  performs an  $H_2$  oracle query to obtain  $(s_i, H_2(PID_i))$ , randomly generates  $x_{i1} \in R_q$  and  $x_{i2} \in R_q$ , computes  $P_i = x_{i1} + x_{i2}P$ , returns  $pk_i = (P_i, H_2(PID_i))$  to  $\mathcal{A}$ , and updates  $L_k$ .

Signcryption query. When  $\mathcal{A}$  queries the input  $(m_i, PID_i)$ ,  $\mathcal{C}$  first searches  $L_k$  according to  $PID_i$  to obtain  $sk_i$  and  $pk_i$  in order to generate a valid signature. Then,  $\mathcal{C}$  randomly selects  $(z_{i1}, z_{i2}, z_{i3}, z_{i4}, w_i) \in R_q$  and  $y_i \in R_q$  and computes

$$t_i = \begin{bmatrix} z_{i1}P + z_{i2} \\ z_{i3}P + z_{i4} \end{bmatrix} - \begin{bmatrix} H_3(PID_i) \\ P_i \end{bmatrix} * w_i. \quad (9)$$

It randomly selects  $c_i \leftarrow \{0, 1\}^n$  and updates  $(t_i, c_i, w_i)$  in the list  $L_5$ . If the tuple  $(t_i, c_i, w_i)$  does not exist in  $L_5$  before the query,  $\mathcal{C}$  returns  $(y_i, c_i, w_i, z_i)$  to  $\mathcal{A}$ ; otherwise, it returns  $\perp$ .

Forgery phase. After a series of queries,  $\mathcal{A}$  outputs a valid forged ciphertext  $(y_i^*, c_i^*, w_i^*, z_i^*)$  for  $(m_i^*, PID_i^*)$  with non-negligible probability  $\varepsilon$ .

After receiving the forged ciphertext  $(y_i^*, c_i^*, w_i^*, z_i^*)$ ,  $\mathcal{C}$  outputs another forged signature  $(y_i', c_i', w_i', z_i')$  for  $(m_i^*, PID_i^*)$  through the forking lemma [38] with non-negligible probability, where  $w_i' \neq w_i^*$ . According to the signature verification algorithm and the collision-resistance property of the hash function, we obtain

$$\begin{bmatrix} z_{i1}^*P + z_{i2}^* \\ z_{i3}^*P + z_{i4}^* \end{bmatrix} - \begin{bmatrix} s_{i1}^* + s_{i2}^*P \\ x_{i1}^* + x_{i2}^*P \end{bmatrix} * w^* = \begin{bmatrix} z_{i1}'P + z_{i2}' \\ z_{i3}'P + z_{i4}' \end{bmatrix} - \begin{bmatrix} s_{i1}^* + s_{i2}^*P \\ x_{i1}^* + x_{i2}^*P \end{bmatrix} * w'. \quad (10)$$

By further converting it into the formulation of the RSIS instance, we obtain

$$[(z_{i1}^* - z_{i1}') + s_{i2}^*(w' - w^*)]P + [(z_{i2}^* - z_{i2}') + s_{i1}^*(w' - w^*)] = 0. \quad (11)$$

According to Lemmas 1 and 2,  $\|z_{i1}^*\|, \|z_{i2}^*\|, \|z_{i1}'\|, \|z_{i2}'\|$  are bounded by  $2\sigma\sqrt{n}$  with overwhelming probability, the following inequality holds:

$$\|(z_{i1}^* - z_{i1}') + s_{i2}^*(w' - w^*)\| \leq \|z_{i1}^* - z_{i1}'\| + \|s_{i2}^*\| \|w' - w^*\| \leq (4\sigma + 2\kappa s)\sqrt{n}. \quad (12)$$

Similarly, it can be obtained that  $\|(z_{i2}^* - z_{i2}') + s_{i1}^*(w' - w^*)\| \leq (4\sigma + 2\kappa s)\sqrt{n}$ .

According to the minimum property, there exists another private key pair  $(s_{i1}', s_{i2}')$  such that  $(s_{i1}', s_{i2}') \neq (s_{i1}^*, s_{i2}^*)$  and  $s_{i1}' + s_{i2}'P = H_2(PID_i^*)$ . Therefore, the following inequality can be derived:

$$[(z_{i1}^* - z_{i1}') + s_{i2}^*(w' - w^*)] - [(z_{i1}^* - z_{i1}') + s_{i2}'(w' - w^*)] = (s_{i2}^* - s_{i2}')(w' - w^*) \neq 0, \quad (13)$$

which means that  $(z_{i1}^* - z_{i1}') + s_{i2}^*(w' - w^*)$  and  $(z_{i1}^* - z_{i1}') + s_{i2}'(w' - w^*)$  cannot both be zero. Similarly, there also exists  $(z_{i2}^* - z_{i2}') + s_{i1}^*(w' - w^*)$  and  $(z_{i2}^* - z_{i2}') + s_{i1}'(w' - w^*)$ , which cannot both be zero.

Therefore, the challenger  $\mathcal{C}$  can obtain a nonzero solution to the RSIS problem with non-negligible probability, namely  $((z_{i1}^* - z_{i1}') + s_{i2}^*(w' - w^*), (z_{i2}^* - z_{i2}') + s_{i1}^*(w' - w^*))$ , where  $\beta > (4\sigma + 2\kappa s)\sqrt{2n}$ . In summary, if  $\mathcal{A}$  can forge a valid ciphertext, then  $\mathcal{C}$  can solve the RSIS problem. However, the RSIS problem is considered hard to solve. Therefore, the adversary cannot forge a valid ciphertext, and the proposed scheme satisfies unforgeability.  $\square$

#### 4.3. Unlinkability

In UASC, a dynamic pseudonym and key update mechanism is designed. Specifically,  $PID_i$  is generated by a double-input hash function, whose inputs are the real identity  $RID_i$  of the node and the hash seeds  $\theta$  provided by the TA, respectively. Among them,  $\theta = yP + e$ , where  $y$  and  $e$  are obtained by random sampling. Because  $\theta$  introduces additional uncertainty and satisfies the RLWE security paradigm, the generation of pseudo identity is highly random and unpredictable, making it difficult for adversaries to recover any valid information about the true identity. The pseudo identities and keys generated in different sessions are independent of each other, and attackers cannot establish effective associations between multiple nodes. It can prevent multiple pseudonyms and keys belonging to the same entity from being linked, thereby realizing the unlinkability protection of nodes. Based on the above dynamic pseudonym and key update mechanism, SN can maintain multiple legal identities for different pseudonyms and can switch keys according to communication requirements. This design not only enhances the privacy protection of the real identity of the node, but also reduces the risk of continuous tracking. In addition, under the background of the rapid development of artificial intelligence and big data analysis technology, the traditional static identity and key mechanism becomes more vulnerable. In contrast, the dynamic pseudonym and key update mechanism introduced by UASC can weaken the correlation between different sessions, thus improving the system's resistance to learning attacks and persistent privacy analysis.

## 5. Comparison and Conclusions

### 5.1. Comparison

In this section, we set  $n = 256$ ,  $q = 2^{24} - 1$  and compare the computational overhead and communication overhead with schemes [44,45] to evaluate the protocol-level feasibility of UASC under a fixed lattice parameter setting. The experiments were conducted on a MacBook Air 2024 equipped with an Apple M3 processor and 16 GB of RAM. The programs were run on a Windows 11 system in a Parallels Desktop 26.2.1 virtual machine and compiled using GCC.

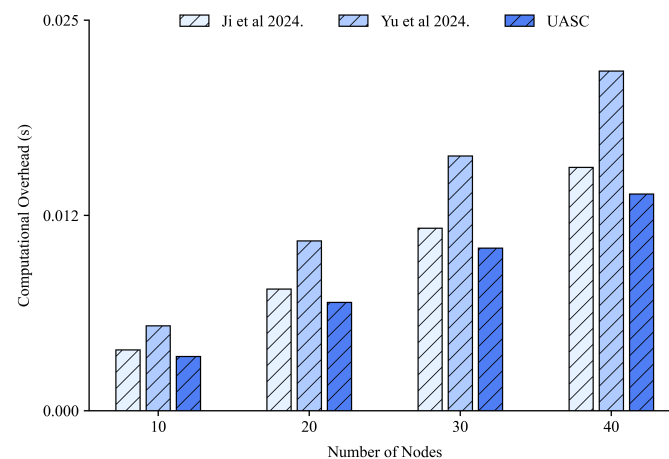
Table 1 compares the security properties of different schemes. Firstly, UASC, scheme [44], and scheme [45] are all constructed based on lattice hard problems. Therefore, they can resist quantum attacks and satisfy the confidentiality requirement of communication data. However, they differ in system design and implementation goals. Specifically, scheme [44] focuses on identity-based data encryption and does not achieve message unforgeability. Secondly, UASC generates part of the private key by using underwater environmental factors in the key generation phase. Scheme [45] adopts a certificateless blind signcryption architecture. Both of them can resist the key escrow problem caused by the central authority. Thirdly, UASC also provides anonymity and unlinkability. To prevent adversaries from impersonating nodes and launching forgery attacks, only pseudonyms are used in communication. This prevents adversaries from identifying the real identities of nodes and forging messages. Meanwhile, to avoid malicious tracking and learning by adversaries, UASC can dynamically update pseudonyms and public–private key pairs. In this way, adversaries cannot link the eavesdropped sessions to the same node and thus cannot collect a large amount of related information to extract sensitive data. Therefore, UASC considers more security risks in UWSNs and provides more comprehensive security properties. It has certain practical significance for real-world applications.

**Table 1.** The comparison of security strength between UASC and other schemes.

| Scheme | Resistance to Key Escrow | Confidentiality | Unforgeability | Anonymity | Unlinkability |
|--------|--------------------------|-----------------|----------------|-----------|---------------|
| [44]   | ×                        | ✓               | ×              | ×         | ×             |
| [45]   | ✓                        | ✓               | ✓              | ×         | ×             |
| UASC   | ✓                        | ✓               | ✓              | ✓         | ✓             |

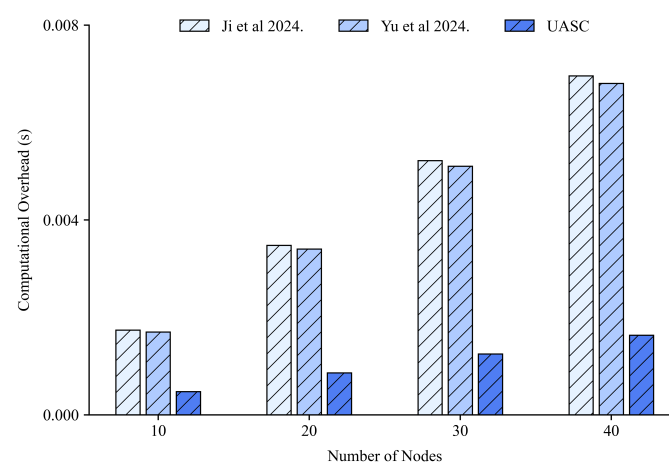
✓: Yes, ×: No

Figure 2 shows the computational overhead of each scheme in the encryption at the sender as the number of nodes increases. As shown in Figure 2, the scheme UASC has less overhead in the encryption stage, which is important for UWSNs. UWSNs are mainly composed of underwater nodes and land servers. The encryption stage of collecting information uploading tasks generally runs on sensor nodes. It is difficult for underwater sensor nodes to obtain energy. The energy consumption speed of nodes determines the service life of nodes, so it is important to reduce node overhead. The scheme [44] designs a Gaussian sampling algorithm on the NTRU lattice trapdoor. Compared with the polynomial single-valued parameters output by the traditional Gaussian sampling, the parameters output by the sampling algorithm are polynomial vectors, which are used to participate in the identity-based encryption system. This process increases the computational complexity of the sender's encryption operation. In order to realize the message blinding function of the scheme, scheme [45] chooses to provide message privacy protection by sacrificing part of the computational overhead and uses more multiplication operations and rejection sampling algorithms. However, UASC only involves polynomial single-valued parameter operations and simultaneously applies signature and encryption to messages by multiplexing the underlying mathematical logic of the algorithm, which reduces the computational overhead, protects the confidentiality of messages and provides the verification ability of message sources.



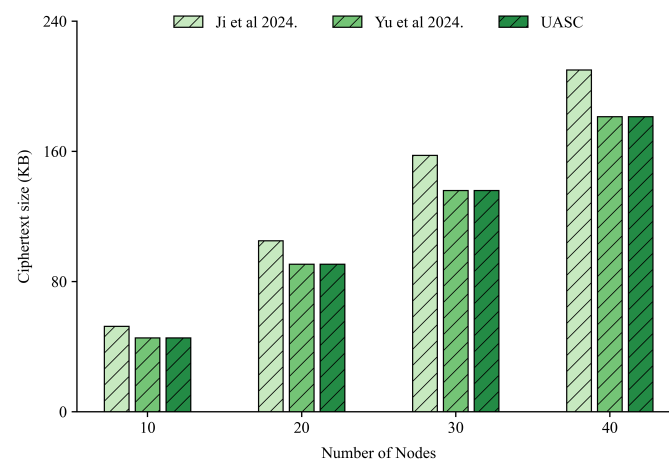
**Figure 2.** The comparison of computational overhead of encryption among UASC, Ji et al 2024's scheme [44] and Yu et al 2024's scheme [45].

Figure 3 compares the computational overhead of different schemes in the batch decryption phase as the number of nodes increases. As shown in Figure 3, UASC has lower computational overhead and a slower growth rate in the batch decryption phase. This is mainly because we use a sink node in the construction of the scheme. The sink node signs the set of valid ciphertexts, so the receiver only needs to authenticate the signature of the sink node when verifying the source of the ciphertext set. By performing single signature verification, the receiver can authenticate the integrity and source legitimacy of the ciphertext set. This not only reduces the computational overhead but also improves the overall response speed of UWSNs. In contrast, when schemes [44,45] process multiple ciphertexts, they need to perform decryption operations for each ciphertext. Compared with scheme [44], scheme [45] uses fewer polynomial multiplication operations in the decryption phase. However, both schemes lack a batch verification mechanism similar to UASC, which can compress the verification of multiple signatures into a single signature verification and avoid the linear growth of verification overhead with the number of nodes. For UWSN scenarios, cooperative operation among nodes is very important. The interaction between a large number of nodes and the server inevitably introduces communication delay. Therefore, by using the sink node in the UWSN structure to design batch verification at the receiver, the response speed can be improved.



**Figure 3.** The comparison of computational overhead of batch decryption among UASC, Ji et al 2024's scheme [44] and Yu et al 2024's scheme [45].

Figure 4 shows the length of the ciphertext generated by the sender in each scheme as the number of nodes increases. For underwater sensor nodes, the main overhead is used for calculation and communication. A longer ciphertext length means more communication overhead. It can be seen that UASC has higher efficiency in transmitting ciphertext, mainly because UASC integrates the process of signature and encryption and reduces the number of parameters generated when each part is independent by reusing some parameters. At the same time, based on the RLWE problem, the transmission parameters are in the form of polynomial single value, which brings lower bandwidth overhead and more compact size than the scheme [44]. Scheme [45] is similar to UASC by constructing polynomial single values on the DNCC problem to resist quantum attacks and reduce communication overhead. The number and types of parameters contained in the ciphertext transmitted by [45] are basically consistent with UASC. Therefore, under the same lattice parameter settings, their ciphertext lengths are theoretically the same. In summary, the design of UASC takes into account the real environment of UWSNs and balances the security and cost overhead of the scheme. While ensuring the low cost overhead of underwater nodes, it provides security against quantum attacks for the transmitted messages and improves the system response rate, so it is more suitable for the application of UWSNs.



**Figure 4.** The comparison of length of the ciphertext among UASC, Ji et al 2024's scheme [44] and Yu et al 2024's scheme [45].

To verify the randomness of the noise-assisted key generation mechanism, MATLAB R2023b simulation experiments were also conducted. The bit disagreement rate between the bit sequences generated by the legitimate node A and the adversary node E, and the Shannon entropy of the bit sequence generated by A were used for evaluation. The results are shown in Table 2. The experimental results show that, under different SNR conditions, the bit disagreement rate between A and E is close to 0.5, and the Shannon entropy of the bit sequence generated by A is close to 1. These results indicate that, under the spatial uniqueness model [46], the bit sequence generated by the noise-assisted mechanism has high randomness and can provide effective environment-related auxiliary randomness for the node key generation process.

**Table 2.** The randomness validation of the noise-assisted key generation mechanism.

| SNR                   | 10 dB   | 15 dB   | 20 dB   | 25 dB   | 30 dB   |
|-----------------------|---------|---------|---------|---------|---------|
| Bit Disagreement Rate | 0.43862 | 0.43065 | 0.42565 | 0.42606 | 0.42472 |
| Shannon Entropy       | 0.99431 | 0.99441 | 0.99447 | 0.99463 | 0.99459 |

## 5.2. Conclusions

In this paper, we propose an underwater aggregate signcryption scheme called UASC on the lattice based on the RLWE and RSIS assumption for the application scenario of collecting data and uploading land servers in UWSNs. UASC reuses the underlying mathematical structure of encryption and signature and aggregates the ciphertext with the help of an intermediate sink node, thereby supporting batch verification at the receiving end to improve the overall communication and processing efficiency of the system. In the Section 3, this paper first introduces the basic structure of UWSNs and the functional positioning of each entity and clarifies the security communication goal of UASC according to the actual needs. Then, the construction method and operation mechanism of UASC are described in detail. In the Section 4, the ciphertext confidentiality and signature unforgeability of UASC are proved by formal proof. In the Section 5, UASC is compared with related schemes in terms of computational overhead and communication overhead. The design ideas and implementation effects of UASC to reduce system overhead are analyzed. Through theoretical and experimental analysis, it is shown that UASC is more suitable for the practical application environment of UWSNs. In future work, we will further improve our scheme in the following aspects. Firstly, we will combine standard security estimation methods for lattice-based cryptography to conduct a more systematic analysis of security, correctness, and performance overhead under different parameter sets and security levels. Second, we will further consider the key characteristics of practical UWSNs, including long propagation delay, limited acoustic bandwidth, packet loss, Doppler frequency shift, node mobility, energy constraints, and time-varying multipath channels, so as to strengthen the connection between UASC and practical UWSN scenarios. Meanwhile, we will further conduct experimental validation by combining underwater network simulations and real underwater acoustic channel data, including key generation randomness, communication delay, energy consumption, and packet loss. In addition, under the premise of ensuring security and correctness, further compressing the ciphertext length and optimizing parameter selection will also be important for future research.

**Author Contributions:** Conceptualization, Z.Z. and M.X.; Methodology, Z.Z. and M.X.; Writing—original draft preparation, Z.Z.; Writing—review and editing, Z.Z. and M.X.; Supervision, M.X.; Funding Acquisition, M.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** Covert Secret Key Agreement Scheme Based on the Underwater Acoustic Dynamic Hypergraph. This work was supported by the National Natural Science Foundation of China under Grant 62172269.

**Data Availability Statement:** Data are contained within the article.

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

## References

1. Fan, R.; Boukerche, A.; Pan, P.; Jin, Z.; Su, Y. An underwater secure localization scheme based on physical layer cryptographic learning. *IEEE Trans. Mob. Comput.* **2026**, *25*, 37–53.
2. Wang, S.; Nguyen, T.L.; Shin, Y. Data collection strategy for magnetic induction based monitoring in underwater sensor networks. *IEEE Access* **2018**, *6*, 43644–43653.
3. Su, R.; Zhang, D.; Li, C.; Gong, Z.; Venkatesan, R.; Jiang, F. Localization and data collection in AUV-aided underwater sensor networks: Challenges and opportunities. *IEEE Netw.* **2019**, *33*, 86–93.
4. Jiang, B.; Feng, J.; Cui, X.; Wang, J.; Liu, Y.; Song, H. Security and reliability of internet of underwater things: Architecture, challenges, and opportunities. *ACM Comput. Surv.* **2024**, *57*, 1–37.
5. Yang, J.; Wu, Z.; Huang, Y.; Gong, M.; Zhang, X.; Ma, Y.; Han, G. K-means++-based secure and efficient routing protocol design for underwater sensor networks. *IEEE Internet Things J.* **2025**, *13*, 1702–1717.

6. 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.
7. Diamant, R.; Casari, P.; Ardizzon, F.; Tomasin, S.; Sherlock, B.; Corner, T.; Neasham, J.A. Channel-based key generation for secure underwater acoustic communications. *IEEE Trans. Wireless Commun.* **2025**, *24*, 5678–5693.
8. Huang, Z.; Peng, X.; Song, Z.; Zhang, B.; Hui, J.; Wu, F.; Jiang, J.; Zhang, Y. High-speed bio-inspired covert underwater acoustic communication technology for the underwater internet of things. *IEEE Internet Things J.* **2025**, *12*, 49737–49753.
9. Fan, R.; Boukerche, A.; Pan, P.; Jin, Z.; Su, Y.; Dou, F. Secure localization for underwater wireless sensor networks via AUV cooperative beamforming with reinforcement learning. *IEEE Trans. Mob. Comput.* **2024**, *24*, 924–938.
10. Jiang, J.; Tian, W.; Han, G.; Zhang, F. A medium access control protocol based on parity group-graph coloring for underwater AUV-aided data collection. *IEEE Internet Things J.* **2023**, *11*, 5967–5979.
11. Donegà, F.; Francescon, R.; Colella, P.; Campagnaro, F.; Nissen, I.; Zorzi, M. A collaborative system for reputation-based security in underwater acoustic networks. *IEEE Trans. Commun.* **2026**, *74*, 4235–4250.
12. Kumar, C.M.; Amin, R.; Brindha, M. SafeCom: Robust mutual authentication and session key sharing protocol for underwater wireless sensor networks. *J. Syst. Archit.* **2022**, *130*, 102650.
13. Zhou, Z.; Gupta, B.B.; Gaurav, A.; Li, Y.; Lytras, M.D.; Nedjah, N. An efficient and secure identity-based signature system for underwater green transport system. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 16161–16169.
14. Shi, J.; Wu, J.; Zhao, Z.; Qi, X.; Zhang, W.; Qiao, G.; Zuo, D. A lightweight secure scheme for underwater wireless acoustic network. *J. Mar. Sci. Eng.* **2024**, *12*, 831.
15. Rupa, C.; Varshitha, G.S.; Divya, D.; Gadekallu, T.R.; Srivastava, G. A novel and robust authentication protocol for secure underwater communication systems. *IEEE Internet Things J.* **2025**, *12*, 47519–47531.
16. Li, X.; Zhu, R.; Du, D.; Jiang, C.; Zhou, Z. ECC-based certificateless aggregate signcryption scheme in cyber-physical power systems. *IEEE Syst. J.* **2024**, *18*, 893–904.
17. Zhang, J.; Dong, C.; Liu, Y. Efficient pairing-free certificateless signcryption scheme for secure data transmission in IoMT. *IEEE Internet Things J.* **2023**, *11*, 4348–4361.
18. Hou, Y.; Cao, Y.; Xiong, H.; Sun, Z.; Mumtaz, S.; Tian, D. CASKA-CRT: Chinese remainder theorem empowered certificateless aggregate signcryption scheme with key agreement in IoVs. *IEEE Trans. Intell. Veh.* **2024**, *9*, 6814–6829.
19. Han, G.; Ma, Y.; Zhang, Z.; Liu, J. Blockchain-assisted efficient certificateless aggregate signcryption scheme for IoMT. *IEEE Internet Things J.* **2026**, *13*, 8452–8464.
20. Gao, Y.; Deng, L.; Feng, S.; Liu, H.; Li, B.; Wang, N. Revocable certificate-based broadcast signcryption scheme for edge-enabled IIoT. *Inf. Sci.* **2025**, *690*, 121540.
21. Shor, P.W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.* **1999**, *41*, 303–332.
22. Kim, J.; Park, J.H. NTRU+: Compact construction of NTRU using simple encoding method. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 4760–4774.
23. Prajapat, S.; Gautam, D.; Kumar, P.; Jangirala, S.; Das, A.K.; Park, Y.; Lorenz, P. Secure lattice-based aggregate signature scheme for vehicular ad hoc networks. *IEEE Trans. Veh. Technol.* **2024**, *73*, 12370–12384.
24. Xie, J.; Hu, Y.; Gao, J.; Gao, W.; Jiang, M. Efficient certificateless signature scheme on NTRU lattice. *KSII Trans. Internet Inf. Syst.* **2016**, *10*, 5190–5208.
25. Li, F.; Bin Muhaya, F.T.Z.; Khan, M.K.; Takagi, T. Lattice-based signcryption. *Concurrency Computat.: Pract. Exp.* **2013**, *25*, 2112–2122.
26. Sourav; Ali, R. Lattice-based ring signcryption scheme for smart healthcare management. *Clust. Comput.* **2024**, *27*, 14131–14148.
27. Sinha, D.; Gupta, S.; Das, I.; Mallick, S.; Tamta, V.P.; Harsha, S.S.; Tiwari, M.; Abdurakhimova, D.; Shandilya, G. Post-quantum identity-based linkable ring signcryption for edge IoT devices. *IEEE Trans. Consum. Electron.* **2026**, *72*, 1876–1889.
28. Prajapat, S.; Kumar, D.; Kumar, P.; Das, A.K.; Hossain, M.S. A lattice-based ring signcryption scheme for secure communication in 6G-enabled vehicular ad hoc networks using blockchain. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 23007–23020.
29. Sourav; Ali, R.; Obaidat, M.S. Secure and efficient lattice-based signcryption for blockchain-enabled IoT healthcare. *IEEE Trans. Dependable Secur. Comput.* **2026**, *23*, 356–367.
30. Bai, Y.; He, D.; Yang, Z.; Luo, M.; Peng, C. Efficient module-lattice-based certificateless online/offline signcryption scheme for Internet of Medical Things. *IEEE Internet Things J.* **2025**, *12*, 27350–27363.
31. Chen, X.; Xu, S.; Gao, S.; Guo, Y.; Yiu, S.-M.; Xiao, B. FS-LLRS: Lattice-based linkable ring signature with forward security for cloud-assisted electronic medical records. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 8875–8891.
32. Yu, H.; Li, B.; Hui, W. Signature-based anti-quantum schemes for blockchain-based donation and e-invoice. *IEEE Internet Things J.* **2024**, *11*, 30245–30259.
33. Yu, H.-F.; Dang, N.-Z.; Yang, B. Blockchain-based insurance claim scheme using proxy signcryption from NTRU lattice. *IEEE Sens. J.* **2025**, *25*, 33801–33809.

34. Lyubashevsky, V. Lattice signatures without trapdoors. In *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 738–755.
35. Güneysu, T.; Lyubashevsky, V.; Pöppelmann, T. Practical lattice-based cryptography: A signature scheme for embedded systems. In *Cryptographic Hardware and Embedded Systems—CHES 2012*; Prouff, E.; Schaumont, P., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7428, pp. 530–547.
36. Ducas, L.; Lyubashevsky, V.; Prest, T. Efficient identity-based encryption over NTRU lattices. In *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 22–41.
37. Chen, X.; Huang, J.; Xiao, K.; Li, H.; Huang, Q. A non-interactive identity-based multi-signature scheme on lattices with public key aggregation. *IEEE Trans. Dependable Secur. Comput.* **2025**, *22*, 4189–4199.
38. Pointcheval, D.; Stern, J. Security arguments for digital signatures and blind signatures. *J. Cryptol.* **2000**, *13*, 361–396.
39. Xi, W.; Li, X.Y.; Qian, C.; Han, J.; Tang, S.; Zhao, J.; Zhao, K. KEEP: Fast secret key extraction protocol for D2D communication. In *Proceedings of the 2014 IEEE 22nd International Symposium of Quality of Service (IWQoS)*; IEEE: Piscataway, NJ, USA, 2014; pp. 350–359.
40. Quist, B.T.; Jensen, M.A. Maximization of the channel-based key establishment rate in MIMO systems. *IEEE Trans. Wirel. Commun.* **2015**, *14*, 5565–5573.
41. Goldsmith, A. *Wireless Communications*; Cambridge University Press: Cambridge, UK, 2005.
42. Pan, P.; Su, Y.; Fan, R.; Yang, S. A secret key generation scheme exploiting spatiotemporal acoustic channel characteristics for underwater sensor networks. *IEEE Sens. J.* **2024**, *24*, 31188–31200.
43. Webster, R.J. A random number generator for ocean noise statistics. *IEEE J. Ocean. Eng.* **2002**, *19*, 134–137.
44. Ji, W.; Wang, Z.; Jin, H.; Wang, Q.; Wang, G.; Gu, D. Identity-based encryption from lattices with more compactness in the standard model. In *Proceedings of the International Conference on Selected Areas in Cryptography*; Springer: Cham, Switzerland, 2024; pp. 259–281.
45. Yu, H.; Zhang, Q.; Li, L. Certificateless anti-quantum blind signcryption for e-cash. *J. Ind. Inf. Integr.* **2024**, *40*, 100632.
46. Wallace, J.W.; Sharma, R.K. Automatic secret keys from reciprocal MIMO wireless channels: Measurement and analysis. *IEEE Trans. Inf. Forensics Secur.* **2010**, *5*, 381–392.

**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.