applied sciences

MDPI

Article

PUF and Chaotic Map-Based Authentication Protocol for Underwater Acoustic Networks

Qi Xie * and Ye Yao

Key Laboratory of Cryptography of Zhejiang Province, Hangzhou Normal University, Hangzhou 311121, China; y.yao@stu.hznu.edu.cn

* Correspondence: qixie68@126.com

Abstract: A secure and effective authentication and communication scheme between users and underwater sensors plays an important role in improving the detection and utilization of marine resources in underwater acoustic networks (UANs). However, due to the energy limitations and susceptibility to capture of underwater sensors and gateways, it is necessary to design a lightweight authentication protocol that can resist capture of sensors and gateways during attacks. In this paper, a lightweight authentication protocol for UANs based on the Physical Unclonable Function (PUF) and chaotic map is proposed. We used the advantages of PUF to resist sensors and gateways being captured in attacks and the chaotic map to achieve lightweight authentication because the computational cost of the chaotic map is almost one-third that of Elliptic Curve Cryptography (ECC). Additionally, we used the formal security proof in the random oracle model to prove the security of the proposed scheme. Our scheme was more secure and efficient compared with some other related schemes in terms of security and performance requirements, and the proposed scheme is suitable for UANs.

Keywords: underwater acoustic networks; authentication; protocol; chaotic maps; PUF

Citation: Xie, Q.; Yao, Y. PUF and Chaotic Map-Based Authentication Protocol for Underwater Acoustic Networks. *Appl. Sci.* **2024**, *14*, 5400. https://doi.org/10.3390/app14135400

Academic Editor: Grzegorz Kołaczek

Received: 17 May 2024
Revised: 16 June 2024
Accepted: 18 June 2024
Published: 21 June 2024

1. Introduction

The ocean area accounts for about 71% of the earth's surface area. With the increasingly prominent contradiction between the global shortage of food, resources, and energy supply and the rapid population growth, the development of marine resources is inevitable for historical development. Using wireless sensor networks (WSNs) to perceive and monitor marine environment information can improve the utilization efficiency of marine resources, coordinate the allocation of marine and land resources, and realize the maximum utilization value of marine resources.

Due to the poor propagation of electromagnetic waves in seawater and since optical communications will be strongly affected by scattering, acoustic waves can enable communications over long-range links, so they provide the most obvious medium to enable underwater communication. Underwater wireless sensor networks are wireless communication networks based on acoustic signals, in which sensors are deployed underwater, where the environments are time-varying, called underwater acoustic networks (UANs). They use aircraft, submarines, or surface ships to randomly deploy a large number of cheap, miniature sensor nodes in the seawater. The nodes form a multi-hop self-organizing network system through underwater acoustic communication, which can cooperatively sense, collect, and process the information of the sensing objects in the network coverage area, and send it to the receiver. They are mainly used to carry out coordinated tasks, such as oceanographic data collection, pollution prediction, ocean mining, shipwreck avoidance, ocean monitoring, etc.

With the increasing use of UANs in industry and the military, the need to transmit sensitive information on insecure channels is also increasing. It is easy for adversaries to

eavesdrop, intercept, modify, and delete the information, which leads to various attacks and causes huge losses [1]. Therefore, it is essential to control access to UANs' information and services and ensure that sensitive information is securely exchanged between users and sensor nodes. At the same time, the UANs are required to be able to respond to the relevant information of the marine environment in real time, reflecting the real-time requirements of the UANs. Only by continuous and real-time monitoring of the changing state of the ocean can humans grasp the ocean data in time, develop, and use the data.

Authentication can ascertain the user legitimacy of using the network resource and establishing the session key between the user and the sensor node to protect the confidentiality and integrity of the data from the attacker. A number of security authentication and key agreement schemes have been proposed for terrestrial wireless sensor networks (TWSNs), but most of them are not applicable to UANs, due to the energy limitations and susceptibility to capture of underwater sensors and gateways. Therefore, a security mechanism specifically for UANs is needed [2].

1.1. Related Work

In 2019, Banerjee et al. [3] proposed a security-enhanced authentication and key agreement scheme for WSN, but their scheme cannot resist offline password guessing attacks, impersonation attacks, and does not achieve session key secrecy, identity unlinkability, and perfect forward secrecy. In 2020, Chen et al. [4] proposed an authentication scheme for WSN in IoT environments, but their scheme is vulnerable to offline password guessing attacks, impersonation attacks, and fails to achieve perfect forward secrecy, user anonymity, and unlinkability. In 2021, Shuai et al. [5] presented a lightweight authentication protocol for WSN environments using ECC to prevent various security issues. However, their scheme does not provide perfect forward security and suffers from desynchronization attacks and stolen-verifier attacks. Later, Kaur et al. [6] presented a two-factor user authentication protocol for smart homes using ECC. Yu et al. [7] presented that Kaur et al.'s scheme cannot resist impersonation attacks, session key disclosure attacks, and secure user authentication. They proposed a lightweight authentication scheme to overcome the security problems of Kaur et al.'s protocol. In 2021, Far et al. [8] proposed a user authentication protocol using fuzzy extractor and hash-chain in the IIoT environment. In 2023, Sahoo et al. [9] proposed a three-factor-based authentication scheme of 5G WSN for IoT systems and claimed that their scheme is secure. However, Xie et al. [10] pointed out that their scheme is vulnerable to user impersonation attacks, sensor node impersonation attacks, and capture attacks, and lacks user unlinkability and three-factor secrecy.

Recently, chaotic map has been widely concerned since it has better security and performance than traditional cryptography. The difficulty of the chaotic map's Diffie–Hellman problem and its semi-group property make it feasible to establish secure session keys. In addition, the computation overhead of a Chebyshev polynomial is approximately 1/3 of the scalar multiplication on elliptic curves [11]. It significantly reduces the computing overhead and energy consumption of resource-constrained sensor nodes, which is more suitable for devices with a limited battery life and smaller computation power. In 2015, Lee et al. [12] proposed a three-party authenticated key agreement scheme based on chaotic maps without a password table. Jabbari et al. [13] showed that the scheme of Lee et al. fails to guarantee user anonymity and put forward an improved scheme. In 2016, Kumari et al. [14] introduced a two-factor authentication scheme for WSN using the chaotic map. However, the protocol of Kumari et al. suffers from sensor node impersonation attacks [15]. In 2018, Aghili et al. [16] proposed an efficient three-factor authentication scheme for WSN using the hash function. However, Wang et al. [17] showed that the scheme does not provide security against session key disclosure attacks, desynchronization attacks, sensor node impersonation attacks, and session-specific temporary information attacks. Besides, they presented an improvement protocol for WSN using chaotic maps. In 2019, Lee et al. [18] introduced a multi-server authentication protocol using extended chaotic maps. However, Kumar et al. [19] found that their protocol is insecure against user impersonation attacks,

session-specific temporary information attacks, and time synchronization problems, and proposed another protocol based on extended chaotic maps. In 2021, Qi et al. [20] proposed a chaotic map-based authentication protocol for an industrial medical cyber-physical system. However, Ding et al. [21] showed that their protocol is vulnerable to identity guessing attacks, user impersonation attacks, trace attacks, desynchronization attacks, and lacks perfect forward secrecy, and they proposed a security-enhanced one.

Recently, how to resist capture attacks from physical devices has become a hot topic in authentication protocol research. Thanks to the application of Physically Unclonable Functions (PUF), many security authentication protocols have emerged that resist sensor capture attacks. In 2024, Xie et al. [22] proposed a multi-server authentication protocol based on PUF and chaotic maps to address the security issues of Yu et al.'s scheme [23]. Xie et al. [24] also proposed a PUF-based security authentication protocol to address the inability of Kumar et al.'s scheme [25] to resist capture attacks from roadside units. Oláh et al. [26] proposed a Blockchain- and PUF-based registration protocol for the Internet of Drones.

The change from hash-based operations to complex cryptographic primitive-based schemes greatly improved the security of TWSNs. However, the difference between TWSNs and UANs makes it impossible to directly use TWSN's secure authentication mechanism for UANs. In 2019, Diamant et al. [27] proposed a cooperative authentication scheme for UANs, which relies on trusted nodes that independently assist in aggregating nodes during the authentication process. Later, Zhang et al. [28] presented a remote mutual authentication scheme based on chaotic maps for UANs. Based on the architecture of underwater wireless sensor networks, Kumar et al. [29] designed an authentication technique that establishes a session key for safe communication. In 2024, Tomović et al. [30] proposed a Blockchain-based Key Management Protocol for UANs, and Wang et al. [31] proposed a deep learning and random forest algorithm-based dynamic trust model for UANs.

1.2. Motivation and Contributions

It is shown that Zhang et al.'s scheme [28] cannot provide secure mutual authentication and establish the session key, and it fails to resist offline password guessing attacks and user impersonation attacks. In Kumar et al.'s scheme [29], the session key between the user and the Onshore Base Station cannot achieve perfect forward secrecy and may suffer from ID guessing attacks. On the other hand, their scheme cannot resist sensor node capture attacks and does not establish the session key between the user and the sensor node. Tomović et al.'s scheme [30] cannot resist sensor node capture attacks and cannot achieve anonymity.

Since almost all authentication protocols for UANs have one or more security flaws, designing a secure and effective lightweight authentication protocol for UANs is a challenge. Therefore, a secure and efficient lightweight authentication protocol for UANs is proposed, and the main contributions are as follows:

(1) Based on the uniqueness and randomness of Physical Uncontrollable Functions (PUF) and the fast computation of chaotic maps, a secure and efficient authentication protocol for UANs is proposed.

(2) The proposed scheme is proven secure under the random oracle model, which can achieve all known security properties, such as perfect forward secrecy, anonymity, and resistance to device capture attacks.

(3) The proposed scheme is more secure and efficient compared with some other related schemes in terms of security and performance requirements, and the proposed scheme is suitable for UANs.

The rest of this paper is constructed as follows: Section 2 provides the preliminaries and the threat model. The proposed authentication and key agreement scheme for UANs is presented in Section 3. Sections 4 and 5 provide corresponding formal and informal analyses of the proposed scheme. The security and performance comparisons between the proposed scheme and other resource-constrained schemes are presented in Section 6. Section 7 is the conclusion.

2. Preliminaries

In this section, we will introduce the threat model used in this paper and review some basic definitions concerning the Chebyshev polynomial, chaotic maps, and PUF.

2.1. Threat Model

The proposed protocol adopted the widely accepted Dolev–Yao threat model (DY model) [32], in which any adversary has the ability to eavesdrop, intercept, modify, or delete the messages transmitted among users, gateways, and sensors. In addition, any adversary can extract all the sensitive information stored in the lost/stolen smart card of a legal user, U_i, using the side channel attack. Meanwhile, any adversary can capture the gateway and sensor nodes.

2.2. Chebyshev Polynomial

Definition 1 (Chebyshev polynomial): *The Chebyshev polynomial can be defined as (1) or (2), where* $n \in N, n \geq 2, T_0(x) = 1, T_1(x) = x$*:*

$$T_n(x) = cos(n \cdot arccos(x)), x \in [-1, 1] \tag{1}$$

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), n \geq 2 \tag{2}$$

The semi-group property and chaos property are two primary properties of the Chebyshev polynomial [33].

Definition 2 (semi-group property): *The semi-group property of the Chebyshev polynomial* $T_n(x)$ *is defined as follows:*

$$\begin{gathered} T_g(T_h(x)) = cos(g \cdot arccos(cos(h \cdot arccos(x)))) \\ T_h(T_g(x)) = cos(h \cdot arccos(cos(g \cdot arccos(x)))) \\ T_g(T_h(x)) = T_h(T_g(x)) \end{gathered} \tag{3}$$

where g and h are positive integers and $x \in [-1, 1]$.

Definition 3 (chaos property): *The Chebyshev polynomial map,* $T_n(x) : [-1, 1] \to [-1, 1]$ *of degree* $n > 1$*, is a chaotic map with invariant density as:* $\partial = \frac{1}{\pi\sqrt{1-x^2}}$*, for the Lyapunov exponent* $\lambda = lnn$.

Definition 4: *Enhanced Chebyshev polynomial is expressed as:*

$$T_n(x) = 2xT_n(x) - T_{n-2}(x) \ mod \ q \tag{4}$$

where q is a large prime and $x \in (-\infty, +\infty)$. The enhanced chaotic maps still satisfy the semi-group property and chaos property.

Definition 5 (chaotic map-based discrete logarithm problem, CMDLP): *Considering* α *and* β*, it is computationally infeasible to compute an integer,* λ*, such that* $T_\lambda(\alpha) \ mod \ q = \beta$.

Definition 6 (chaotic map-based Diffie–Hellman problem, CMDHP): *Considering* α*,* $T_\lambda(\alpha)$*, and* $T_\beta(\alpha)$*, it is computationally infeasible to compute* $T_{\lambda\beta}(\alpha)$.

2.3. Physically Unclonable Functions

As a new hardware security primitive, the Physically Unclonable Function (PUF) is a hardware function implementation circuit that relies on chip features, with uniqueness and randomness. By extracting process parameter deviations that are inevitably introduced during chip manufacturing, it achieves a function that uniquely corresponds to the excita-

tion and response signals [34]. In our scheme, PUF was used to protect the information stored in the gateway and sensors.

3. The Proposed Scheme

Based on the fact that gateways and sensors in underwater acoustic networks are easily captured, the proposed scheme adopted PUF to protect the secret information stored in gateways and sensors. In order to achieve two-factor security, the user's identity and password are verified using fuzzy authentication. To achieve lightweight and secure authentication, the semi-group property of Chebyshev polynomials was adopted to achieve perfect forward secrecy. The notations used in our scheme are listed in Table 1.

Table 1. Notations.

Symbol	Description
U_i	The i^{th} user
GWN_k	The k^{th} gateway node
Sn_j	The j^{th} sensor node
SC	Smart card of user U_i
id_i, pw_i	Identity and password of U_i, respectively
Gid_k	Identity of the k^{th} gateway node
S_K	Secret key of the GWN_k
Sid_j	Identity of the j^{th} sensor node
S_j	Secret key of the Sn_j
rug, ru	Two random numbers selected by U_i
r	The random number selected by GWN_k
T_i	Timestamp
ΔT	The allowable maximum transmission time interval
$M_1 \parallel M_2$	Data M_1 concatenates with data M_2
$M_1 \bigoplus M_2$	XOR operation of M_1 and M_2
$h(.)$	A secure one-way hash function
$E_k(.)/D_k(.)$	Symmetric encryption/decryption using the key k

3.1. Initialization Phase

The initialization phase is executed offline by the gateway. Gateway GWN_k randomly chooses S_K as its master key, Gid_k as its identity, a large prime number q, and a secure one-way hash function $h(.)$. Meanwhile, the gateway chooses a challenge C_G, and computes the corresponding response, $R_G = PUF1(C_G)$, $GSK = S_K \bigoplus h(R_G)$, or $M_0 = T_{S_K}(x)\ mod\ q$. Then, the gateway chooses $x \in (-\infty, +\infty)$, and publishes the public parameters $\{q, x, h(.)\}$. In the same way, the gateway chooses Sid_j as the identity of the sensor node Sn_j, and computes $S_j = h(Sid_j \parallel Gid_k \parallel S_K)$ according to the topological relationship for the sensor node Sn_j. The gateway sends $\{Gid_k, Sid_j, S_j\}$ to Sn_j, and stores $\{M_0, Gid_k, C_G, Sid_j, PUF1(), GSK\}$.

The sensor node Sn_j chooses a challenge C_S, and computes the corresponding response $R_S = PUF2(C_S)$ and $SNJ = S_j \bigoplus h(R_S)$, and stores $\{Gid_k, Sid_j, SNJ, C_S, PUF2()\}$.

3.2. User Registration Phase

The user performs the following steps to be a legal user through a secure channel.

Step 1: The user U_i freely selects the identity id_i and password pw_i, and sends the registration request message $\{id_i, h(id_i \parallel pw_i)\}$ to the GWN_k through a secure channel.

Step 2: After receiving the registration request message, the GWN_k computes $M_1 = h(id_i \parallel S_K) \bigoplus h(id_i \parallel pw_i)$, where $x \in (-\infty, +\infty)$. Then, the GWN_k stores $\{h(.), M_0, M_1\}$ in a smart card (SC) and safely issues the SC to the user U_i.

Step 3: The user U_i computes $M_3 = h(id_i \parallel M_0 \parallel pw_i) mod n_0$, where $n_0 \in (2^4, 2^8)$, and stores $\{h(.), M_0, M_1, M_3, n_0\}$

3.3. Login Phase

In order to login to the GWN_k and access the data from the Sn_j, the user U_i needs to execute the following steps:

After inserting the SC into the card reader of a specific terminal device, U_i enters its identity id_i and password pw_i, computes $M_3' = h(id_i \parallel M_0 \parallel pw_i) mod\ n_0$, and checks whether $M_3' = M_3$ is correct or not. If yes, the SC generates two random numbers, rug and ru, and computes $M_2 = M_1 \bigoplus h(id_i \parallel pw_i)$, $K_U = T_{ru}(x) mod\ q$, $K_{U-G} = T_{ru}(M_0)\ mod\ q$, $M_4 = h(T_1 \parallel M_2 \parallel rug \parallel id_i)$, and $M_5 = E_{K_{U-G}}\left(id_i \parallel Sid_j \parallel rug \parallel M_4\right)$, where T_1 is the current timestamp. Then, the SC sends the login request message, $\{K_U, M_5, T_1\}$, to the GWN_k.

3.4. Authentication and Key Management Phase

This phase allows the user to accomplish mutual authentication and session key agreement between the user and the sensor node through the help of the gateway node, and the steps are described as follows.

Step 1: After receiving the login request message, $\{K_U, M_5, T_1\}$, the GWN_k first computes whether $T_2 - T_1 \leq \Delta T$ holds, where T_2 is the current timestamp. If the timestamp verification holds, GWN_k continues to execute the next step, otherwise, the login request is denied.

Step 2: The GWN_k computes $K_{U-G} = T_{S_K}(K_U)\ mod\ q$, $M_6 = D_{K_{U-G}}(M_5) = \{id_i, Sid_j, rug, M_4\}$.

Step 3: The GWN_k computes $S_K = GSK \bigoplus h(PUF1(C_G))$, $M_2^{'} = h(id_i \parallel S_K)$, $M_4^{'} = h\left(T_1 \parallel M_2^{'} \parallel rug \parallel id_i\right)$.

Step 4: The GWN_k checks whether $M_4^{'} = M_4$ is correct or not. If the equation holds, the U_i and GWN_k are successfully authenticated by each other, otherwise, GWN_k terminates this session instantaneously.

Step 5: The GWN_k chooses a random nonce r, computes $S_j = h(Sid_j \parallel Gid_k \parallel S_K)$, $M_7 = E_{S_j}(h(id_i \parallel r) \parallel Gid_k \parallel Sid_j \parallel K_U \parallel T_2)$, and delivers the message $\{M_7, T_2\}$ to the sensor node Sn_j.

Step 6: Upon obtaining the message $\{M_7, T_2\}$ at timestamp T_3, the Sn_j checks whether $T_3 - T_2 \leq \Delta T$ is correct. If it holds, they move to the next step, otherwise, this session is terminated instantaneously.

Step 7: The Sn_j computes $S_j = NJ \bigoplus h(PUF2(C_S))$, $M_8 = D_{S_j}(M_7) = \{h(id_i \parallel r), Gid_k, Sid_j, K_U\}$.

Step 8: The Sn_j chooses a random number, rs, and computes $M_9 = T_{rs}(x)\ mod\ q$, $SK = h\left(T_{rs}(K_U) \parallel K_U \parallel M_9 \parallel Gid_k \parallel Sid_j \parallel h(id_i \parallel r)\right)$, $M_{10} = h(Gid_k \parallel Sid_j \parallel SK \parallel M_9 \parallel T_3)$, and $M_{11} = h(S_j \parallel M_9 \parallel M_{10} \parallel T_3 \parallel h(id_i \parallel r))$. Then, the Sn_j delivers the message $\{M_9, M_{10}, M_{11}, T_3\}$ to GWN_k.

Step 9: Upon obtaining the message $\{M_9, M_{10}, M_{11}, T_3\}$ at timestamp T_4, the Sn_j checks whether $T_4 - T_3 \leq \Delta T$ holds. If yes, they move to the next step, otherwise, this session is terminated instantaneously.

Step 10: GWN_k first verifies the correctness of M_{11}, and then computes $M_{12} = h(rug \parallel M_9 \parallel M_{10} \parallel T_4)$, $M_{13} = E_{rug}(r \parallel M_{12})$, and sends $\{M_9, M_{10}, M_{13}, T_3, T_4\}$ to the user U_i.

Step 11: Upon receiving the message $\{M_9, M_{10}, M_{13}, T_3, T_4\}$ at timestamp T_5, the SC checks whether $T_5 - T_4 \leq \Delta T$, decrypts M_{13}, obtains r and M_{12}, and checks the correctness of M_{12}. If yes, they proceed to the next step, otherwise, this session is terminated instantaneously.

Step 12: The SC computes $SK = h(T_{ru}(M_9) \parallel K_U \parallel M_9 \parallel Gid_k \parallel Sid_j \parallel h(id_i \parallel r))$, $M_{10}^{'} = h(Gid_k \parallel Sid_j \parallel SK \parallel M_9 \parallel T_3)$, and checks whether $M_{10}^{'} = M_{10}$ holds. If the equation holds, a session key, SK, is established.

The Login and authentication process is shown in Table 2.

Table 2. Login and authentication phase.

U_i	GWN_k	Sn_j
insert SC and input id_i, pw_i		
RC compute $M_3' = h(id_i \parallel M_0 \parallel pw_i) mod\ n_0$ Checks whether $M_3' = M_3$		
$M_2 = M_1 \oplus h(id_i \parallel pw_i)$		
choose two random numbers rug and ru		
$K_U = T_{ru}(x) mod\ q$		
$K_{U-G} = T_{ru}(M_0) mod\ q$		
$M_4 = h(T_1 \parallel M_2 \parallel rug \parallel id_i)$		
$M_5 = E_{K_{U-G}}(id_i \parallel Sid_j \parallel rug \parallel M_4)$ $\xrightarrow{\{K_U,\ M_5,\ T_1\}}$		
	check the freshness of T_1	
	compute $K_{U-G} = T_{S_K}(K_U) mod\ q$	
	$M_6 = D_{K_{U-G}}(M_5)$	
	$S_K = GSK \oplus h(PUF1(C_G))$	
	$M_2' = h(id_i \parallel S_K)$	
	$M_4' = h\left(T_1 \parallel M_2' \parallel rug \parallel id_i\right)$	
	check whether $M_4' = M_4$	
	$S_j = h(Sid_j \parallel Gid_k \parallel S_K)$	
	choose a random number r	
	$M_7 = E_{S_j}(h(id_i \parallel r) \parallel Gid_k \parallel Sid_j \parallel K_U \parallel T_2)$ $\xrightarrow{\{,M_7,T_2\}}$	check the freshness of T_2
		$S_j = NJ \oplus h(PUF2(C_S))$
		$M_8 = D_{S_j}(M_7)$
		choose a random number rs
		$M_9 = T_{rs}(x) mod\ q$
		$SK = h(T_{rs}(K_U) \parallel K_U \parallel M_9 \parallel Gid_k \parallel Sid_j \parallel h(id_i \parallel r))$
		$M_{10} = h(Gid_k \parallel Sid_j \parallel SK \parallel M_9 \parallel T_3)$
		$M_{11} = h(S_j \parallel M_9 \parallel M_{10} \parallel T_3 \parallel h(id_i \parallel r))$
	check the freshness of T_3	$\xleftarrow{\{\{\ M_9\ ,\ M_{10}\ ,\ M_{11}\ ,\ T_3\ \}}$
	verify the correctness of M_{11}	
	compute $M_{12} = h(rug \parallel M_9 \parallel M_{10} \parallel T_4)$	
	$M_{13} = E_{rug}(r \parallel M_{12})$ $\xleftarrow{\{\ M_9\ ,\ M_{10}\ ,\ M_{13}\ ,\ T_3\ ,T_4\ \}}$	
check the freshness of T_4		
decrypt$M_{13}$and obtain rand M_{12}		
verify the correctness of M_{12}		
$SK = h(T_{ru}(M_9) \parallel K_U \parallel M_9 \parallel Gid_k \parallel Sid_j \parallel h(id_i \parallel r))$		
$M_{10}' = h(Gid_k \parallel Sid_j \parallel SK \parallel M_9 \parallel T_3)$		
check $M_{10}' = M_{10}$		
if it holds, the session key is SK.		

3.5. Password Update Phase

For the security consideration, a legal user should be allowed to update the personal password. In this phase, when the user wants to update his password, pw_i, to a new password, pw_i^{new}, the user needs to enter his identity, id_i, old password, pw_i, and new password, pw_i^{new}, after inserting the SC into the card reader. The SC computes $M_3' = h(id_i \parallel M_0 \parallel pw_i) mod\ n_0$, and checks whether $M_3' = M_3$ is correct or not. If yes, the SC computes $M_1^{new} = M_1 \oplus h(id_i \parallel pw_i) \oplus h\left(id_i \parallel pw_i^{new}\right)$ and replaces M_1 with M_1^{new}.

4. Formal Security Analysis

This section will formally analyze the security of the proposed scheme. The results demonstrated that our scheme was proven secure. The notions of the model used in this paper are defined as follows:

Participants: In the proposed scheme, $\mathcal{P}$, denoted as P, the participants include the user U, the gateway GW, and the sensor node Sn. In the i^{th} instance, the participants, the user, the gateway, and the sensor node are denoted as INS_P^i, INS_U^i, INS_{GW}^i, and INS_{Sn}^i, respectively.

States of Oracle: Oracle in our scheme has three states: $ACCEPT$, $REJECT$, and $\perp$. If an oracle receives a correct request message, the state is $ACCEPT$, if the request message is illegal, the state is $REJECT$. When the above conditions do not occur, the state is $\perp$.

We defined that if the oracle INS_U^i (INS_{Sn}^i) is $ACCEPT$, and the session key SK_U^i (SK_{SN}^i) has been negotiated with INS_P^i, then INS_U^i (INS_{Sn}^i) obtains its session identity SID_U^i (SID_{Sn}^i), and the corresponding participant identity PID_{Sn}^i (PID_U^i).

Partnering: If the following conditions are satisfied, INS_U^i and INS_{Sn}^i are $ACCEPT$, and the session key has been negotiated, then INS_U^i and INS_{Sn}^i are considered as partners.

1. The session key SK_U^i generated by INS_U^i equals INS_{Sn}^i's session key, SK_{SN}^i.
2. INS_U^i and INS_{Sn}^i are in the same session; that is, $SID_U^i = SID_{Sn}^i$.
3. The participant identities of INS_U^i and INS_{Sn}^i are equal to INS_{Sn}^i and INS_U^i, respectively.

Queries: To simulate multiple attacks, queries are defined as follows:

Execute (INS_P^i): Execute simulates the eavesdropping attack, and A executes this query to obtain all the transcripts.

Send (INS_P^i, *message*): This query simulates the sending operation executed by the adversary, A. The message is sent to oracle INS_P^i; if the message is correct, INS_P^i responds to A based on $\mathcal{P}$, otherwise, the message is neglected.

Reveal (INS_P^i): If the session key has been negotiated, INS_{Sn}^i and INS_U^i are in $ACCEPT$, and the query Test has not been executed yet. The query Reveal will reveal the session key when it is executed. Otherwise, the output is null.

Corrupt (INS_U^i): This query simulates a corruption attack. It will return the message $\{h(.), M_0, M_1\}$ to the adversary, which is stored in the smart card.

Test (INS_P^i): This query is allowed to be executed at most once. The query generates a random bit r; if $r = 1$ and the session key has been generated, the session key is sent to the adversary. Otherwise, A receives a random number.

Freshness: An instance INS_P^i can be identified as fresh if it satisfies the following conditions:

1. Reveal query has not been executed.
2. Corrupt is executed at most once.
3. INS_{Sn}^i and INS_U^i are in $ACCEPT$.

Semantic Security: As the definition of the Test query shows, r determines if the output is the session key. Furthermore, A generates a random bit r_a; if $r_a = r$, A knows the correctness. The possibility is $Adv_{\mathcal{P}}^A = |2\Pr[r = r_a] - 1| = |2\Pr[suc(A)] - 1|$. If $Adv_{\mathcal{P}}^A \geq \eta$, $\mathcal{P}$ is not secure, where η is sufficiently small.

CMDLP: The chaotic map-based discrete logarithm problem (CMDLP) is distributed as: considering $x, y \in Z_p^*$, where $y = T_r(x) mod\ p$. Computing r is computationally hard. The advantage of CMDLP is $Adv_A^{CMDLP} = 2\Pr\left[A(x, y) = r : r \in Z_p^*, y = T_r(x) mod\ p\right]$.

Theorem 1. Assume A is the adversary that tries to break $\mathcal{P}$ in PPT. A is allowed to execute multiple Execute and Send queries. The Test query is permitted to execute at most once. We identified q_{se}, q_h, q_{Send}, and q_{Exe} as the execute numbers of symmetric encryption, hash operation, Send, and Execute queries, respectively. l_{se}, l_h, n, and l_{pw} are the lengths of the output of symmetric encryption, hash operation, transcript, and password, respectively. The advantage of breaking $\mathcal{P}$ by A in PPT is:

$$Adv_{\mathcal{P}}^{A} \leq \frac{q_{se}^2}{2^{l_{se}}} + \frac{q_h^2}{2^{l_h}} + \frac{(q_{Send} + q_{Exe})^2}{n} + \frac{q_{Send}}{2^{l_{pw}-1}} + 2Adv_{A}^{CMDLP}$$

Proof. We assume that the adversary A tends to break the scheme $\mathcal{P}$ in the probabilistic polynomial time (PPT). Meanwhile, we define games, denoted as $Game_i (0 \leq i \leq 4)$, to simulate multiple attacks launched by A. According to $Game_i$, the event $Ev_i (0 \leq i \leq 4)$ represents that A breaks $\mathcal{P}$ in $Game_i$. The games are defined as:

$Game_0$: This game simulates the real attack launched by A. First, A guesses the random bit r; hence, we have:

$$Adv_{\mathcal{P}}^{A} = |2\Pr[Ev_0] - 1| \tag{5}$$

$Game_1$: This game simulates the eavesdropping attack. A executes multiple Execute queries and at most one Test query. After obtaining the output of the Test query, A has to figure out if the output is the session key according to the captured transcripts, $\{K_U, M_5, M_7, M_9, M_{10}, M_{11}, M_{13}, T_i\}$. Here, $K_U = T_{ru}(x) mod\ q$, $M_5 = E_{K_{U-G}}(id_i \parallel Sid_j \parallel rug \parallel M_4)$, $M_7 = E_{S_j}(h(id_i \parallel r) \parallel Gid_k \parallel Sid_j \parallel K_U \parallel T_2)$, $M_9 = T_{rs}(x) mod\ q$, $M_{10} = h(Gid_k \parallel Sid_j \parallel SK \parallel M_9 \parallel T_3)$, $M_{11} = h(S_j \parallel M_9 \parallel M_{10} \parallel T_3 \parallel h(id_i \parallel r))$, $M_{13} = E_{rug}(r \parallel M_{12})$, and timestamps. $SK = h(T_{rs}(K_U) \parallel K_U \parallel M_9 \parallel Gid_k \parallel Sid_j \parallel h(id_i \parallel r)) = h(T_{ru}(M_9) \parallel K_U \parallel M_9 \parallel Gid_k \parallel Sid_j \parallel h(id_i \parallel r))$. This session key is based on CMDLP, and A cannot compute SK according to the messages or figure out the relationship between the session key and the transcripts because the one-way hash function, random numbers, and timestamps are used. Therefore, we have:

$$Pr[Ev_0] = Pr[Ev_1] \tag{6}$$

$Game_2$: This game simulates A and executes the Execute and Send queries to launch the collision attacks among transmitted messages. These messages are symmetric encrypted or hashed. According to the birthday paradox, the probability of collision of the symmetric encryption is $\frac{q_{se}^2}{2^{l_{se}+1}}$. The probability of hash collision is $\frac{q_h^2}{2^{l_h+1}}$. The collision probability of transcripts is $\frac{(q_{send}+q_{exec})^2}{2n}$. Therefore, we have:

$$Pr[Ev_2] - Pr[Ev_1] \leq \frac{q_{se}^2}{2^{l_{se}+1}} + \frac{q_h^2}{2^{l_h+1}} + \frac{(q_{Send} + q_{Exe})^2}{2n} \tag{7}$$

$Game_3$: This game simulates that after the Corrupt query is executed, A launches guessing attacks on the password. A can obtain $\{h(.), M_0, M_1\}$ stored in the smart card. Here, $M_0 = T_{rk}(x)\ mod\ q$ and $M_1 = h(id_i \parallel G_k) \bigoplus h(id_i \parallel pw_i)$. The probability of guessing the password by A is $\frac{1}{2^{l_{pw}}}$; therefore, we have:

$$Pr[Ev_3] - Pr[Ev_2] \leq \frac{q_{Send}}{2^{l_{pw}}} \tag{8}$$

$Game_4$: This game simulates that A calculates SK according to $M_9 = T_{rs}(x) mod\ q$ and $K_U = T_{ru}(x) mod\ q$, which are transmitted openly. According to the definition, we have:

$$Pr[Ev_4] - Pr[Ev_3] \leq Adv_{A}^{CMDLP} \tag{9}$$

The probability of guessing the random bit r is $1/2$, which is equal to the probability of guessing the session key. We have:

$$Pr[Ev_4] = \frac{1}{2} \tag{10}$$

Combining (5) to (10), we have: $\frac{1}{2}Adv_{\mathcal{P}}^{A} \leq \frac{q_{se}^2}{2^{l_{se}+1}} + \frac{q_h^2}{2^{l_h+1}} + \frac{(q_{Send}+q_{Exe})^2}{2n} + \frac{q_{Send}}{2^{l_{pw}}} + Adv_{A}^{CMDLP}$.

That is:

$$Adv_{\mathcal{P}}^{A} \leq \frac{q_{se}^2}{2^{l_{se}}} + \frac{q_h^2}{2^{l_h}} + \frac{(q_{Send}+q_{Exe})^2}{n} + \frac{q_{Send}}{2^{l_{pw}-1}} + 2Adv_{A}^{CMDLP} \quad (11)$$

□

5. Informal Security Analysis

5.1. Offline Password Guessing Attack

Since the information in smart cards can be retrieved by side channel attacks, such as power analysis attacks, stolen smart card attacks should be considered when designing authentication schemes using smart cards. In our scheme, if the SC is stolen by an adversary, it can retrieve the information stored in the SC and eavesdrops on the message transferred on the public channel. Though the adversary can guess the user's identity and password and obtain M_2, he still cannot know the random nonce rug, and can not verify whether $M_4 = h(T_1 \parallel M_2 \parallel rug \parallel id_i)$ is correct or not. Therefore, the adversary cannot know whether his guessed identity and password are correct or not. On the other hand, if an adversary wants to guess id_i and pw_i to satisfy $M_3 = h(id_i \parallel M_0 \parallel pw_i) mod\ n_0$, there are 2^{32} candidates for the (id_i, pw_i) pair when n = 256. Moreover, the adversary cannot know which pair is correct. Thus, our scheme can withstand the stolen smart card attack and offline password guessing attack.

5.2. Mutual Authentication

In our scheme, only the legitimate user with the correct identity and password can pass the verification. In the authentication and key agreement phase, U_i transmits message $\{K_u, M_5, T_1\}$ via the public channel, and only GWN_k can recover the encryption key K_{U-G} to decrypt M_5 and obtain $\{id_i, Sid_j, rug, M_4\}$. If GWN_k verifies M_4 successfully, the user can authenticate GWN_k by checking the correctness of M_{12}, so our scheme achieves mutual authentication between U_i and GWN_k. In the same way, GWN_k transmits the encrypted data M_7 to the sensor node, and only the sensor node can decrypt the message and verify the correctness of GWN_k to achieve mutual authentication between GWN_k and Sn_j. Thus, it could provide mutual authentication among the user, the gateway, and the sensor node.

5.3. User Impersonation Attack

In our scheme, if an adversary wants to impersonate the user, he must know the message, M_2, which can verify the legitimacy of the user. However, M_2 is protected by the user's identity and password, and the adversary cannot verify whether his guessed identity and password are correct or not. Therefore, our scheme can withstand the impersonation attack.

5.4. Man-in-the-Middle Attack

An adversary, A, could intercept messages transferred on a public channel. In our scheme, an adversary, A, needs to make the GWN_k believe that it is from the user, U_i. However, the adversary, A, cannot pass the verification without the identity, id_i, and password, pw_i, to calculate M_2. Meanwhile, only the GWN_k can calculate K_{U-G} to decrypt M_5 and encrypted messages with the encryption key S_j to the sensor node Sn_j, so the adversary cannot impersonate the user and the gateway node. In the same way, the adversary cannot impersonate the sensor node since the adversary does not know S_j to decrypt the encrypted message. Therefore, the scheme can withstand the man-in-the-middle attack successfully.

5.5. Malicious Insider Attack

If a malicious insider attacker can impersonate a user, U_i, he must know $h(id_i \parallel S_k)$ of the user U_i. In our scheme, the U_i's password is protected by the collision-resistant one-way hash function $h(.)$, and according to the analysis in Section 5.1, the adversary cannot obtain pw_i and $h(id_i \parallel pw_i)$. Therefore, the attacker cannot compute $h(id_i \parallel S_k)$ from M_1. Meanwhile, it cannot obtain $h(id_i \parallel S_k)$ from the gateway node and the sensor node. Therefore, our scheme can withstand the malicious insider attack.

5.6. Replay Attack

In our scheme, we used timestamp and the random number to resist replay attacks. In each session of the scheme, random numbers, ru, rs, and rug, are generated by the user and the sensor node to establish the session keys, and the session keys of each session are calculated relying on these random numbers. Meanwhile, these messages are protected by the encryption algorithm and hash function. Therefore, our scheme can withstand the replay attack.

5.7. Perfect Forward Secrecy

This secrecy means that the disclosure of a long-term master key will not lead to past session key disclosure. In the proposed scheme, if the GWN_k's long-term private key, S_k, is leaked to the attacker, it does not help the adversary to reveal the past session keys. The session key is computed as $SK = h\big(T_{ru}(M_9) \parallel K_U \parallel M_9 \parallel Gid_j \parallel Sid_j \parallel h(id_i \parallel r)\big) = h\big(T_{rs}(K_U) \parallel K_U \parallel M_9 \parallel Gid_j \parallel Sid_j \parallel h(id_i \parallel r)\big)$. The parameters ru and rs are generated randomly and uniquely for every session. Meanwhile, it is computationally infeasible to compute $T_{rs}(T_{ru}(x))$ according to $T_{rs}(x)$ and $T_{ru}(x)$ due to the hardness of CMDHP. Therefore, our scheme can achieve perfect forward secrecy.

5.8. Known Session Key Attack

If the implementation of the authentication scheme can generate a unique session key, and the compromise of the key has no effect on other session keys, the authentication scheme can provide known session key security. In the proposed scheme, the session key, SK, is unique to each session run because the random numbers ru and rs are generated randomly and independently by the user and the sensor node. Therefore, our scheme can provide known session key security.

5.9. Anonymity and Non-Traceability

Our scheme provides user anonymity, as an adversary cannot obtain or eavesdrop on the user identity, id_i, in the login and authentication phase because the identity, id_i, is transferred in encrypted form by an encryption key K_{U-G} and GWN_k is a trusted entity. Meanwhile, the encryption key is generated randomly for every new session, so the message is dynamic for each session, and it is unable to distinguish between different users. Therefore, our scheme achieves user anonymity and cannot be traced.

5.10. Immunity from Bergamo et al.'s Attack

If both $T_s(x)$ and x are known, then one can determine $s^{'}$, such that $T_{s'}(x) = T_s(x)$. More precisely, $s^{'} = \frac{arccos(T_s(x))+2k\pi}{arccos(x)} fork \in Z^+$. However, this attack cannot happen according to the paper of Zhang et al. [33], because Bergamo et al.'s attack [35] is based on the value range x $\in [-1, 1]$. Our proposed scheme uses the enhanced Chebyshev polynomial, $T_n(x) = 2xT_n(x) - T_{n-2}(x)\ mod\ q$, where q is a large prime and $x \in (-\infty, +\infty)$, so our proposed scheme can avoid Bergamo et al.'s attack.

5.11. Sensor Node and Gateway Capture Attacks

In the proposed scheme, all sensor nodes, Sn_j, and gateways, GWN_k, are deployed with PUF to protect the stored secret information, so our scheme can resist sensor node and gateway capture attacks.

6. Performance Comparison

This section will analyze and compare the proposed scheme with other related schemes [5,7,8,13,28–30] in terms of security and computation costs, which are presented in Tables 3 and 4.

Table 3. Comparison of computation costs.

	User (A)	Gateway Node/Server	Sensor Node/User (B)	Total	Execution Cost
[5]	$7T_h + 2T_s$	$4T_h$	$10T_h + 2T_s$	$21T_h + 4T_s$	3.668 ms
[7]	$11T_h + 1T_s + 1T_f$	$11T_h$	$7T_h$	$29T_h + 1T_s + 1T_f$	10.57 ms
[8]	$9T_h + 2T_e + 1T_f$	$10T_h + 1T_e$	$5T_h$	$24T_h + 3T_e + 1T_f$	33.784 ms
[13]	$4T_h + 4T_c + 2T_s$	$4T_h + 4T_c + 4T_s$	$4T_h + 3T_c + 2T_s$	$12T_h + 11T_c + 8T_s$	39.22 ms
[28]	$11T_h + 3T_c + 1T_s$	$11T_h + 3T_c + 2T_s$	$5T_h + 3T_c + 1T_s$	$27T_h + 9T_c + 4T_s$	31.832 ms
[29]	$5T_h$	$5T_h + T_m$	$3T_h + T_m$	$13T_h + 2T_m$	33.036 ms
[30]	-	$1T_h + 3T_e + 2T_s$	$1T_h + 2T_e + T_s$	$2T_h + 5T_e + 3T_s$	42.006 ms
Our scheme	$6T_h + 3T_c + 2T_s$	$5T_h + 1T_c + 3T_s$	$3T_h + 2T_c + 1T_s$	$14T_h + 6T_c + 6T_s$	22.816 ms

Table 4. Comparison of security features.

	[5]	[7]	[8]	[13]	[28]	[29]	[30]	Our Scheme
Anonymity	T	T	F	T	T	T	F	T
Non-traceability	T	T	T	T	T	T	T	T
Mutual authentication	T	T	T	T	F	F	T	T
Resist malicious insider attack	T	T	T	T	F	T	T	T
Resist offline password guessing attack	T	T	T	T	F	T	T	T
Resist stolen smart card attack	T	T	T	F	F	T	T	T
Resist replay attack	T	T	F	F	T	T	T	T
Resist impersonation attack	F	F	F	F	F	T	T	T
Perfect forward secrecy	F	F	F	T	T	F	T	T
Known session key secrecy	T	T	T	T	T	F	T	T
Resist sensor node capture attack	F	F	F	T	F	F	F	T

The client program is written based on JAVA and deployed on a mobile phone, with the environment (Version: Android 13, Hardware: MediaTek Dimensity 8100, 8GB of RAM, Mali-G610 MC6 GPU), and the cryptographic operations are based on JAC library. The server program is written based on Python and deployed on the Ubuntu virtual machine (Version: 22.04.3 LTS, Hardware: 64-bit AMD 860K CPU @ 3.7GHz 8GB RAM), and the cryptographic operations are based on the gmpy2 library and pycrypto library. The sensor program is written based on Python and deployed on the Raspberry Pi 4B (Broadcom BCM2711, 1.5 GHz, 64-bit, ARM Cortex-A72, RAM: 2GB LPDDR4-3200 RAM). According to the requirements of the protocol, the interaction at the registration stage is based on a secure channel, so we used the WebSocket library to construct the secure channel. WebSocket is a protocol that enables full-duplex communication over a single TCP connection and supports TLS. The interaction at the authentication stage is based on an open channel and is implemented using sendto in the WebSocket library. Sendto directly sends data based on UDP, which has higher efficiency compared to TCP.

All the above devices were tested under the WIFI 1000 Mbps environment. We tested the transmission and reception delay during the registration and authentication, respectively. Here, we took the average value in the relevant schemes. Taking 512-bit

data in the TLS channel in the registration stage and 2048-bit data in the open channel in the authentication stage as examples, a total of 1000 tests were conducted to obtain the average values. Table 5 shows the test results. The measured results indicated that the time overhead for a single transmission and reception was on the microsecond level (1 µs $= 10^{-3}$ ms $= 10^{-6}$ s). The transmission delay was much lower than the hardware operation, so in the analysis of time complexity, we ignored the transmission delay.

Table 5. Transmission delay.

	Registration (512 Bits, TLS)	Verification (2048 Bits, Public)
Phone Server	11.382 µs	7.142 µs
Sensor Server	13.451 µs	7.129 µs
Phone Sensor	-	6.974 µs

Since the time for computing the XOR operation and string concatenation could be ignored, as compared with other cryptographic primitive-based operations, we only considered the time to calculate the one-way hash function (T_h), deterministic reproduction function of fuzzy extractor (T_f), Chebyshev chaotic map polynomial (T_c), elliptic curve point multiplication (T_e), modular multiplication (T_m), and symmetric encryption/decryption (T_s). In the environment of Windows 7 64-bit AMD 860K CPU @ 3.7GHz 8GB RAM, the computational times were approximately 0.068 ms, 8.038 ms, 3.084 ms, 8.038 ms, 16.076 ms, and 0.56 ms, respectively.

From Tables 3–5, we can see that our scheme had a lower computation cost and higher security.

7. Conclusions

Few lightweight, underwater acoustic network authentication schemes have been designed due to the change in the data transmission environment and propagation medium. Thus, this work proposed a lightweight authentication and key agreement scheme for UANs, which adopted PUF to protect the secret information stored in the gateway and sensors, used the fuzzy verifier to achieve two-factor secrecy, and used the semi-group property of Chebyshev polynomials to achieve lightweight authentication and perfect forward secrecy. We used the widely accepted formal security proof in the random oracle model to prove the security of our scheme. Compared to existing schemes, the proposed protocol had higher security and improved the computational efficiency by 39.52% compared to the best existing solutions, with perfect forward security. As a result, the proposed scheme is efficient and more suitable for battery-powered devices in the underwater acoustic networks.

Author Contributions: Q.X., resources, writing—review and editing, supervision, funding acquisition; Y.Y., formal analysis, writing—original draft. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Hangzhou Joint Fund of the Zhejiang Provincial Natural Science Foundation of China (Grant No. LHZSZ24F020002) and the National Natural Science Foundation of China (Grant No. U21A20466).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors have no conflicts of interest/competing interests to disclose.

References

1. Xie, Q.; Ding, Z.; Xie, Q.; Tan, X.; He, D.; Tang, W. Blockchain-based traffic accident handling protocol without third-party for VANETs. *IEEE Internet Things J.* 2024, *early access*. [CrossRef]
2. Domingo, M.C. Securing underwater wireless communication networks. *IEEE Trans. Wirel. Commun.* **2011**, *18*, 22–28. [CrossRef]
3. Banerjee, S.; Chunka, C.; Sen, S.; Goswami, R.S. An enhanced and secure biometric based user authentication scheme in wireless sensor networks using smart cards. *Wirel. Pers. Commun.* **2019**, *107*, 243–270. [CrossRef]
4. Chen, C.T.; Lee, C.C.; Lin, I.C. Efficient and secure three-party mutual authentication key agreement scheme for WSNs in IoT environments. *PLoS ONE* **2020**, *15*, e0232277.
5. Shuai, M.; Yu, N.; Wang, H.; Xiong, L.; Li, Y. A Lightweight Three-Factor Anonymous Authentication Scheme With Privacy Protection for Personalized Healthcare Applications. *J. Organ. End. User Com.* **2021**, *33*, 1–18. [CrossRef]
6. Kaur, D.; Kumar, D. Cryptanalysis and improvement of a two-factor user authentication scheme for smart home. *J. Inf. Secur. Appl.* **2021**, *58*, 102787. [CrossRef]
7. Yu, S.; Jho, N.; Park, Y. Lightweight Three-Factor-Based Privacy-Preserving Authentication Scheme for IoT-Enabled Smart Homes. *IEEE Access* **2021**, *9*, 126186–126197. [CrossRef]
8. Far, H.A.N.; Bayat, M.; Das, A.K.; Fotouhi, M.; Pournaghi, S.M.; Doostari, M.A. LAPTAS: Lightweight anonymous privacy-preserving three-factor authentication scheme for WSN-based IIoT. *Wirel. Netw.* **2021**, *27*, 1389–1412.
9. Sahoo, S.S.; Mohanty, S.; Sahoo, K.S.; Daneshmand, M.; Gandomi, A.H. A Three Factor based Authentication Scheme of 5G Wireless Sensor Networks for IoT System. *IEEE Internet Things* **2023**, *10*, 15087–15099. [CrossRef]
10. Xie, Q.Y.; Xie, Q. Security Analysis on a Three-Factor Authentication Scheme of 5G Wireless Sensor Networks for IoT System. *IEEE Internet Things* **2024**, *11*, 15038–15042. [CrossRef]
11. He, D.; Kumar, N.; Lee, J.-H.; Sherratt, R.S. Enhanced three-factor security protocol for consumer USB mass storage devices. *IEEE Trans. Consum. Electr.* **2014**, *60*, 30–37.
12. Lee, C.-C.; Li, C.-T.; Chiu, S.-T.; Lai, Y.-M. A new three-party-authenticated key agreement scheme based on chaotic maps without password table. *Nonlinear Dynam* **2015**, *79*, 2485–2495. [CrossRef]
13. Jabbari, A.; Mohasefi, J. Improvement in new three-party-authenticated key agreement scheme based on chaotic maps without password table. *Nonlinear Dynam* **2019**, *95*, 3177–3191. [CrossRef]
14. Kumari, S.; Li, X.; Wu, F.; Das, A.K.; Arshad, H.; Khan, M.K. A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps. *Future Gener. Comp. Syst.* **2016**, *63*, 56–75. [CrossRef]
15. Li, J.; Zhang, W.; Kumari, S.; Choo, K.K.R.; Hogrefe, D. Security analysis and improvement of a mutual authentication and key agreement solution for wireless sensor networks using chaotic maps. *Trans. Emerg. Telecommun. Technol.* **2018**, *29*, e3295. [CrossRef]
16. Aghili, S.F.; Mala, H.; Peris-Lopez, P. Securing heterogeneous wireless sensor networks: Breaking and fixing a three-factor authentication protocol. *Sensors* **2018**, *18*, 3663. [CrossRef] [PubMed]
17. Wang, F.; Xu, G.; Xu, G. A provably secure anonymous biometrics-based authentication scheme for wireless sensor networks using chaotic map. *IEEE Access* **2019**, *7*, 101596–101608. [CrossRef]
18. Lee, T.-F.; Diao, Y.-Y.; Hsieh, Y.-P. A ticket-based multi-server biometric authentication scheme using extended chaotic maps for telecare medical information systems. *Multimed. Tools Appl.* **2019**, *78*, 31649–31672. [CrossRef]
19. Kumar, A.; Om, H. An enhanced and provably secure authentication protocol using Chebyshev chaotic maps for multi-server environment. *Multimed. Tools Appl.* **2021**, *80*, 14163–14189. [CrossRef]
20. Qi, R.; Ji, S.; Shen, J.; Vijayakumar, P.; Kumar, N. Security preservation in industrial medical CPS using Chebyshev map: An AI approach. *Future Gener. Comp. Syst.* **2021**, *122*, 52–62. [CrossRef]
21. Ding, Z.; Xie, Q. Provably secure and lightweight three-factor authentication scheme for industrial medical CPS. *J. Inf. Secur. Appl.* **2023**, *79*, 103656. [CrossRef]
22. Xie, Q.; Zhao, Y. Physical Unclonable Function based Lightweight Three-factor Authentication for Multi-Server Architectures. *Mathematics* **2024**, *12*, 79. [CrossRef]
23. Yu, Y.; Taylor, O.; Li, R.; Sunagawa, B. An Extended Chaotic Map-Based Authentication and Key Agreement Scheme for Multi-Server Environment. *Mathematics* **2021**, *9*, 798. [CrossRef]
24. Xie, Q.; Huang, J. Improvement of a Conditional Privacy-Preserving and Desynchronization-Resistant Authentication Protocol for IoV. *Appl. Sci.* **2024**, *14*, 2451. [CrossRef]
25. Kumar, P.; Om, H. A conditional privacy-preserving and desynchronization-resistant authentication protocol for vehicular ad hoc network. *J. Supercomput.* **2022**, *78*, 17657–17688. [CrossRef]
26. Oláh, N.; Molnár, B.; Huszti, A. Secure Registration Protocol for the Internet of Drones Using Blockchain and Physical Unclonable Function Technology. *Symmetry* **2023**, *15*, 1886. [CrossRef]
27. Diamant, R.; Casari, P.; Tomasin, S. Cooperative Authentication in Underwater Acoustic Sensor Networks. *IEEE Trans. Wirel. Commun.* **2019**, *2*, 954–968. [CrossRef]
28. Zhang, S.; Du, X.; Liu, X. A secure remote mutual authentication scheme based on chaotic map for underwater acoustic networks. *IEEE Access* **2020**, *8*, 48285–48298. [CrossRef]
29. Kumar, C.M.; Amin, B.; Brindha, M. SafeCom: Robust mutual authentication and session key sharing protocol for underwater wireless sensor networks. *J. Syst. Archit.* **2022**, *130*, 102650. [CrossRef]

30. Tomović, S.; Krivokapić, B.; Nađ, D.; Radusinović, I. BEKMP: A Blockchain-Enabled Key Management Protocol for Underwater Acoustic Sensor Networks. *IEEE Access* **2024**, *12*, 74108–74125. [CrossRef]
31. Wang, B.; Yue, X.; Liu, Y.; Hao, K.; Li, Z.; Zhao, X. A Dynamic Trust Model for Underwater Sensor Networks Fusing Deep Reinforcement Learning and Random Forest Algorithm. *Appl. Sci.* **2024**, *14*, 3374. [CrossRef]
32. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
33. Zhang, L. Cryptanalysis of the public key encryption based on multiple chaotic systems. *Chaos Soliton Fract.* **2008**, *37*, 669–674. [CrossRef]
34. Aman, M.N.; Chua, K.C.; Sikdar, B. Mutual Authentication in IoT Systems Using Physical Unclonable Functions. *IEEE Internet Things J.* **2017**, *4*, 1327–1340. [CrossRef]
35. Bergamo, P.; D'Arco, P.; De Santis, A.; Kocarev, L. Security of public-key cryptosystems based on Chebyshev polynomials. *IEEE Trans. Circuits* **2005**, *52*, 1382–1393. [CrossRef]