Designing Anonymous Signature-Based Identity Authentication Scheme for Ocean Multilevel Transmission

Jun Ye, Xinhui Cao, and Shaoxiong Xie

***Abstract*—With the rapid development of exploration in the ocean, identity authentication technologies have been applied in marine data transfer environments to address the challenges of security and privacy. However, sending data directly to the target server is often challenging during the process of offshore data transmission. Therefore, relay nodes are utilized to assist the data transmission. To guarantee data security in this situation, a multilevel data transmission identity authentication protocol (AIAS-oceanMT) is introduced in this article, which is tailored for the complex network conditions in the marine environment. Based on elliptical curve and digital signature technologies, two stages of authentication as identity authentication and data authentication are offered. During data processing, only XOR operations, hash computations, and elliptic curve cryptography (ECC) operations are used, also, efficiency and security are effectively balanced. The security of our protocol is supported by a real-or-random (ROR) model. Furthermore, lower communication and computational overheads is demonstrated from our protocol in comparative analyses on security feature with other protocols, which is confirmed by the simulate experiment.**

***Index Terms*—Anonymist, authentication, elliptic curve cryptography (ECC), ocean security, provable security.**

I. Introduction

THE RAPID development of the new generation of information technology has promoted a high-level integration of big data, the Internet of Things, satellite communications, artificial intelligence, and many technologies with real industries. At the same time, it has also brought more problems to marine information security. This brings challenges to marine informatization and digitalization, which also brings opportunities for constructing a new marine information security system.

On the one hand, building a "digital ocean" can effectively manage and organize marine information resources and use them in a more scientific, comprehensive, shared, and sustainable manner to provide a strong guarantee for marine informatization. An effective supportive tool can be provided to offer essential data and basic functions for marine scientific research and management.

On the other hand, all devices need to go through the authentication process before communicating with each other so that the identity authentication will be one of the crucial technologies to ensure the security of marine information. According to the original intention of marine communication networks, devices can communicate and share data. If devices communicate without identity authentication, important information may be stolen by network attacks, which affects the security of the system.

Traditional authentication models designed over the past few decades cannot provide complete network protection. Network attacks can be categorized into seven types as masquerading attacks, man-in-the-middle attacks, denial-of-service attacks, forgery attacks, guessing attacks, physical attacks, and routing attacks. Different from other attacks on fake identities, the attacker of masquerade attacks are would simulate the identity of a legitimate user. Masquerade attacks can be divided based on their method how the attacker attempts to imitate an existing component or system, where attackers can also act like legitimate users. Amin et al. [1] explained how to protect the network from user simulated attacks during identity authentication in their protocol. Wallrabenstein [2] proposed a device authentication method using physical unclonable function (PUF). Ganta et al. [3] discussed synthetic attacks in auxiliary information, and Baig et al. [4] showed the way to prevent synthetic attacks in noninteractive data publishing environments by combining sampling and generation. Additionally, an adversary can steal identity authentication data from the server during a current or past identity authentication session. Then, the adversary can attempt to gain entry into the server with leaked data.

A. Motivation

Marine networks typically consist of multiple devices as sensors, fixed offshore platforms, satellites, and fixed land platforms. These constitute a complex marine network environment, which leads to the complexity of marine data transmission and makes it difficult to ensure data security and privacy. Additionally, traditional identity authentication technology can not address these issues due to the limited resource of devices. The devices collecting marine information are generally far away from land so that the transmitting

Manuscript received 6 January 2024; revised 29 February 2024 and 29 March 2024; accepted 15 April 2024. Date of publication 25 April 2024; date of current version 26 June 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62162020, and in part by the Science Project of Hainan University under Grant KYQD(ZR)20021. *(Corresponding author: Jun Ye.)*

The authors are with the School of Cyberspace Security and the Key Laboratory of Internet Information Retrieval of Hainan Province, Hainan University, Haikou 570228, Hainan, China (e-mail: yejun@hainanu.edu.cn; 2447156102@qq.com; 197115773@qq.com).

Digital Object Identifier 10.1109/JIOT.2024.3390176

and authenticating of data is hard to complete in one step. Therefore, it is needed to use relay devices for data forwarding. However, the use of relay devices for information transmission is vulnerable to adversary attacks. It may face security risks, such as eavesdropping, tampering, counterfeiting, and replay attacks. Based on the above difficulties, the primary motivation of this article is to address the security issue of long-distance data transmission under limited device resource. Therefore, based on previous identity authentication methods, a multilevel data transmission identity authentication protocol suitable for the marine environment is proposed. With this protocol, data transmission security problems can be addressed in long-distance environments, and the security and privacy of data transmission is better enhanced.

B. Research Contributions

In this article, a multilevel data transmission identity authentication protocol suitable for marine environment was proposed to solve the security problems of device identity authentication and multinode data transmission. Since the data is collected in the ocean, when the collection device sends data out, it is not easy to transmit it all at once so that the help of relay devices for data forwarding is required. Therefore, a multilevel data transmission method is adopted. The collection devices are responsible for collecting and encrypting ocean data. The relay devices are responsible for forwarding the data and verifying the identity of the data source devices. The server is responsible for decrypting the data and verifying identities of devices in the entire data transmission process. This protocol is suitable for multinode transmission networks. During the data transmission process, signature technology is used to ensure the integrity of the data, and authentication technology is used to ensure the authenticity and legality of the devices. The following are the main contributions of our protocol.

1) A multilevel data transmission and identity authentication protocol suitable for the marine environment is proposed and applied to the secure transmission of marine data collection. After each transmission, the identity of the data sender is authenticated to verify the legality and authenticity of the data.
2) While ensuring data security, elliptic curve point multiplication is used more rarely. The experimental comparative analysis shows that this solution effectively reduced the computation overhead of devices.
3) While ensuring efficient communication, this solution can track the entire data transmission process and effectively resist the corruption of a single entity. The performance analysis comparing this scheme to other schemes reveals an effective reduction of communication overhead introduced by this scheme.

C. Paper Organization

The structure of this article is as follows. We will introduce some related research in Section II. In Section III, we will introduce some related knowledge and system overview involved in the protocol. Then, we will focus on our proposed solution in Section IV. Next, the security analysis of the resolution is conducted in Section V, and experimental research and comparison are in Section VI. Finally, our solution is summarized in Section VII.

II. Related Work

With the rapid development of modern networks, applications of networks have entered all walks of life, in which we must consider the reliability of the participants. Authentication is an essential method in reliability assurance, which plays a significant role in the network environment. In recent research, Wei et al. [5] proposed a lightweight authentication key protocol with privacy protection, effectively addressing the issue of too much computational and communication overhead in the AKA scheme. However, this protocol is unsuitable for complex marine network environments considering the long-distance transmission. Comparing with traditional identity authentication methods, Wang et al. [6] proposed a method using biological characteristics for identity authentication, allowing a rapid identity recognition process and effectively improving the overall efficiency of the system. However, during the authentication process, the biometric method can easily lead to the leakage of identity information and is unsuitable for authentication between devices.

For early identity authentication protocols, Amin et al. [7] proposed the key negotiation between users and the cloud server following with an authentication phase employed with bilinear pairing. In the Schemes [8], [9], [10], bilinear pairing is also employed, ensuring user's anonymity during the identity authentication phase. However, bilinear pairing is computationally expensive for resource-constrained devices. In order to alleviate the aforementioned issues, Zhang et al. [11] proposed to store critical data in the device ahead of the authentication stage. However, this method did not solve the problem but only had a mitigating effect.

In order to solve the identity authentication problem of resource-constrained devices, Zheng et al. [12] designed a new lightweight identity authentication protocol based on PUF. This protocol effectively reduces computing and storage overhead and is suitable for point-to-point IoT device authentication. However, this protocol is extremely inefficient in multilevel transmission environments and is nearly not usable in our research environment. Ding et al. [13] designed a lightweight anonymous authentication protocol based on elliptic curve cryptography (ECC) and signature encryption technology, which effectively solves the problem of resource-constrained devices and has low computing and communication overhead. Wazid et al. [14] proposed a remote user identity authentication protocol applied in a smart home environment. This protocol effectively solve the above problems. However, it introduces other issues. It is challenging to prevent disguised adversaries from penetrating the identity authentication process, which leads to the leakage of identity information. A lightweight identity authentication protocol for wearable devices based on hash and XOR operations was designed by Gope and Sikdar [15]. However, this protocol also has some security implications.

TABLE I
COMPARATIVE ANALYSIS BETWEEN THE CONTRIBUTIONS OF THIS ARTICLE AND EXISTING RESULTS

	Resource-constrained environment	Resist key leakage	Securit y proof	Forward Secrecy:	Replay attack	Man-in-the-middle attack	Anonymity
[26]	×	×	√	√	√	×	√
[25]	√	√	√	√	√	×	√
[30]	√	√	√	√	√	√	√
[27]	√	×	√	×	√	√	×
[5]	√	×	√	√	√	√	√
[7]	×	√	×	√	√	×	√
[19]	×	√	√	√	√	√	√

In the Internet of Things environment, various researchers have proposed different identity authentication protocols to address identity authentication challenges in specific contexts. Without considering anonymous authentication, numerous schemes like [16] and [17] have been validated for the authentication of network device identities. Li et al. [18] proposed a password negotiation scheme that does not offer anonymous authentication. Naveed Aman et al. [19] introduced an authentication scheme grounded in dynamic energy tradeoffs, specifying distinct security strategies tailored to the varying security requirements across different environments to minimize resource consumption. For anonymous authentication, Simplicio et al. [20] developed an identity authentication protocol that facilitates anonymity. Shen et al. [21] introduced a lightweight certificate-free anonymous authentication protocol with cloud assistance. However, traditional cloud deployment is remote, which significantly impact computational overhead. Concurrently, bilinear pairing is also an effective method for anonymous authentication, but requiring substantial computational resources. Additionally, group signature technology is a widely utilized anonymous authentication method.

The AKA protocol is a traditional authentication method for securing communications, but it is hard to meet the escalating security requirements. In order to address the authentication challenges faced by resource-constrained devices, Chen et al. [25] introduced a lightweight AKA protocol based on ECC. Throughout the protocol flow, authentication encryption relies only on Hash operations, XOR operations, and the ECC algorithm, achieving a reduced computational overhead. However, the communication overhead of this protocol is relatively high. Lian et al. [26] proposed an authentication protocol for IoT devices that employs power-efficient operations to minimize computational overhead. Nevertheless, the computational overhead of this protocol remains elevated and is not suitable for resource-constrained scenarios comparing to other schemes. Roy et al. [27] put forward an authentication and key exchange protocol for resource-constrained devices, which utilizes computationally efficient operations to ensure secure communication and incorporates the PUF technique to generate device identity credentials to mitigating physical attacks. However, it is limited to authentication between devices in single-hop scenarios.

In order to solve the problem of multilevel data transmission security, Aman et al. [28] proposed a multihop indirect data transmission method to complete the whole communication process with the help of other auxiliary devices, as well as to complete the authentication of identity between devices based on PUF. In order to protect data privacy, Yin et al. [29] adopted fingerprint authentication technique to solve the privacy leakage problem and proposed a lightweight fingerprint template with variable length to help resource-constrained devices with data privacy protection. However, this scheme has limited application scenarios and is difficult to meet the communication authentication between devices. Li et al. [30] proposed a lightweight ECC-based authentication scheme that realizes revocable attributes. This scheme was applied to authenticate between resource-constrained devices in a power grid, achieving a low computational and communication overhead. But the scheme is only applicable to single-hop scenarios and cannot realize the authentication requirements in multihop environments.

For the above party studies, we show the analysis result of some schemes in Table I, containing the features of security characteristics, whether considering resource-constrained environments and whether resisting related security attacks.

Although all schemes above have good ideas on how to realize secure authentication, we have to design a new authentication protocol considering that the scenario of our research is a marine open network environment and the goal of guaranteeing the high efficiency of data transmission as well as the security of data. Therefore, a multilevel data transmission and identity authentication protocol for the marine environment is proposed in order to solve various problems in the practical application environment.

III. PRELIMINARIES

A. System Model

The specific application of this solution is to ensure secure communication between devices while transmitting data in a marine open network environment. The data transmission model is depicted in Fig. 1. This model comprises collection devices, relay devices, and a server.

Collection Device: It possesses a unique identity ID and a corresponding identity certificate, enabling it to perform computing operations prescribed by the scheme.

Relay Device: It holds a pair of public and private keys, along with a unique identity ID. Its public key is exposed.

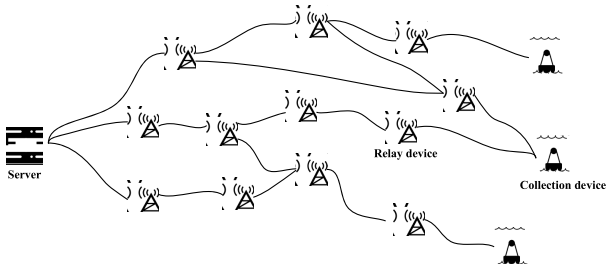


Fig. 1. Marine communication model.

Server: It is trustworthy and secure in this system. Additionally, it is responsible for initializing the public parameters of this system and registering both collection and relay devices.

B. Assumptions

The aim of this study is to address the challenge of marine data transmission security within an open network environment. Owing to the limited environment and device's performance, three entities are defined in this article as data collection devices, relay devices, and the server. Within this model, the server is considered trustworthy and secure. Relay devices are considered secure but they are possible to be corrupted. Data collection devices have limited resources, such as memory, computing power, and responsiveness, whereas the server possesses no such limitations. The placement of data collection equipment relies on the location of relay equipment, which is strategically planned based on the data collection range. Meanwhile, the server is fixed onshore and possesses a wide range of signal reception, large storage space, and strong computation capabilities.

C. Security Requirements

The protocol described in this article must meet the following security requirements.

1) Authentication is required for data transmission between relay devices.
2) Data correctness authentication is required during data transmission through multilevel relay devices.
3) Establishment of information about the data transmission process is needed, including the identity and location of each relay node through which data is transmitted.
4) Must ensure privacy by providing forward secrecy and anonymity.
5) Must ensuring that the information in the memory of each device is unattainable to prevent forgery attacks.

IV. Proposed Protocol

In this section, a multilevel data transmission and identity authentication protocol for the marine environment, which is mainly applied to marine data collection, is described in detail. And the research of this protocol is dedicated to address the issue of oceanic data transmission security. The protocol consists of three main phases: 1) the system initialization phase; 2) the device registration phase; and 3) the authentication and data decryption phase. During the initialization phase, the server sets the system parameters and publishes them. In the registration phase, each collection device and relay node obtains its private key with the assistance of the server. The collection device acquires its identity credentials, while the relay node obtains other data. Finally, in the authentication and data decryption phase, the original data is transmitted to the server through many relay nodes. And server verifies and decrypts the data. Additionally, to resist replay attacks, timestamp authentication of communication messages is employed within the current system. This is a typical assumption suitable for authentication mechanisms in various network environments. The notations listed in Table II and their descriptions are utilized to discuss the phases in the subsequent sections.

TABLE II
Notation Used in This Article

Symbol	Significance
Client	Collection device
RD	Relay device
ΔT	Maximum transmission delay
$\oplus, \|$	Bitwise XOR and concatenation operations,respectively
$E_q(u,v)$	A non-singular elliptic curve : $y^2 = x^3 + ux + v(\text{mod } p)$
P	A base point in $E_q(u,v)$
$x \cdot Q$	An elliptic curve point multiplication $x \in Z_q^*$ $Q \in E_q(u,v)$
K_{pri}/K_{pub}	Public and Private key pair of Server is used for RD $K_{pub} = K_{pri} \cdot P$
$h(\cdot)$	A cryptographic (collision resistant) one-way hash function
(s, T_{pub})	Public and Private key pair of Server is used for client $T_{pub} = s \cdot P$
(d_i, Q_i)	Public and Private key pair of RD $Q_i = d_i \cdot P$
$\{data\}_{key}$	Encrypt data using AES
t_i	Current timestamps
sign	Signature Information
Auth	Authentication information

A. Initialization Phase

The server performs the following steps to select system parameters.

Consider a nonsingular elliptic curve $E_q(u, v)$ on a prime (finite) field $Z_q = \{0, 1, \ldots, q-1\}$ with base point P of the form $y^2 = x^3 + ux + v(\text{mod } p)$.

Next, the server chooses a conflict-resistant one-way hash function with the form $h : \{0,1\}^* \to \{0,1\}^l$, which accepts an input string with arbitrary length and generates a fixed-length output as a message digest (hash value).

The server then chooses a random secret value $s \in Z_q^*$ and computes $T_{\text{pub}} = s \cdot P$. It then chooses a random number $K_{\text{pri}} \in Z_q^*$ as the system private key and computes the corresponding system public key $K_{\text{pub}} = K_{\text{pri}} \cdot P$.

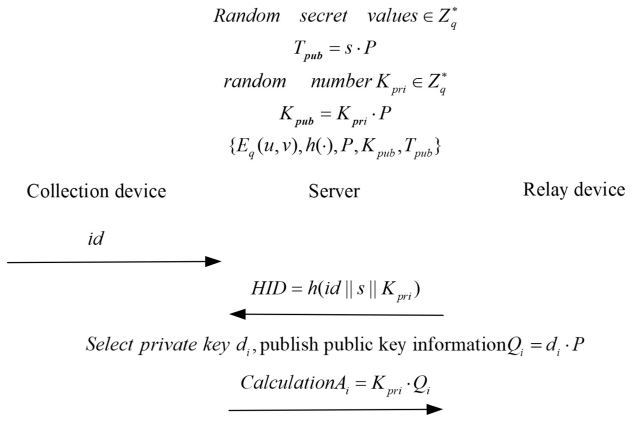


Fig. 2. System registration process.

And then the server publishes the public system parameters $\{E_q(u, v), h(\cdot), P, K_{\text{pub}}, T_{\text{pub}}\}$.

B. Registration Phase

Each collection device or relay device selects its identity (ID) and then sends it to the server through a secure channel. The server generates the identity credentials $HID = h(id||s||K_{\text{pri}})$ for the collection device. At the same time, the server negotiates with the relay device to assist the relay device in generating and saving the private key information d_i, and the public information $Q_i = d_i \cdot P$ is published, and the server generates information $A_i = K_{\text{pri}} \cdot Q_i$. After that, the service sends information HID and $A_i = K_{\text{pri}} \cdot Q_i$ to the collection device and relay device, respectively. The process in detail is shown in Fig. 2.

C. Authentication Phase

Step 1: The collection device selects a random number r$\in Z_q^*$ and obtains the current timestamp t_1. Next, it calculates $R = r \cdot P$, then it generates confusing identity information by calculating $\text{MID} = id \oplus h(r \cdot T_{\text{pub}})$ and the authentication code $\text{MAC} = h(id||\text{HID}||t_1||R)$. At the same time, it calculates the key $= h(\text{HID}||t_1||R)$. Using the key as the public key, they encrypt the collected raw data with a symmetric encryption algorithm (AES) to obtain the ciphertext data $\{\text{data}\}_{\text{key}}$. After that, it sends the message $M = \{t_1, R, \text{MAC}, \text{MID}, \{\text{data}\}_{\text{key}}\}$ to the relay device through the public channel at the time corresponding to the timestamp of t_1.

Step 2: After the relay device receives the request message at time of timestamp t_2, it first judges whether $|t_2 - t_1| < \Delta T$. If the condition is valid, it accepts the data, otherwise it gives up the data received this time and sends a message to the data source device requesting it to resend. The relay device generates a random number $r_i \in Z_q^*$ and computes $N_i = r_i \cdot Q_j$ (where Q_j is the public key of the next device to receive the data), $R_i = r_i \cdot P$ and selects the temporary identity $\text{TID}_i = h(\text{id}_i||r_i)$ and the next timestamp t_3, and then computes $\text{AID}_i = h(N_i) \oplus \text{TID}_i$ and $\text{Auth}_{ij} = h(\text{TID}_i||R_i||t_3||A_iQ_j)$. Next, the relay device computes the signature message $\text{sign}_i = h(M \cdot Q_i) \oplus \text{Auth}_{ij}$ and the data message $M_i = d_i \cdot M \oplus A_iQ_j$. When the data is ready, it sends the data $\{\text{sign}_i, M_i, M, R_i, \text{AID}_i, t_3\}$ over the co-channel to the next level of the relay device at the time of timestamp t_3.

Step 3: After the next level relay device receives the request message at time of timestamp t_4, it first judges whether $|t_4 - t_3| < \Delta T$. If the condition is valid, it accepts the data, otherwise it discards the data received this time and sends a message to the data source device to request it to resend. The current relay device will calculate $N_j = d_j \cdot R_i$, restore TID_i as $\text{TID}_i' = AID \oplus h(N_j)$, and then calculate $\text{Auth}_{ij}' = h(\text{TID}_i'||R_i||t_3||A_jQ_i)$ and $(M_i \oplus A_jQ_i) \cdot P$. To authenticate the identity of former relay device i, relay device j computes $h((M_i \oplus A_jQ_i) \cdot P)$ and compares it with $(\text{sign}_i \oplus \text{Auth}_{ij}')$, if it is equal then the authentication is successful. Next, the later device j also generates the random number $r_j \in Z_q^*$ as shown in the previous step and computes $N_j' = r_j \cdot Q_k$ (where Q_k is the public key of the relay device that will receive the data next) as well as $R_j = r_j \cdot P$ and selects the temporary identity $\text{TID}_j = h(\text{id}_j||r_j)$ and the timestamp t_5 for next sending time. Then $\text{AID}_j = h(N_j') \oplus \text{TID}_j$ and $\text{Auth}_{jk} = h(\text{TID}_j||R_j||t_5||M_jQ_k)$ are computed. Finally, the relay device j computes the signature message $\text{sign}_j = h(M_i \cdot Q_j) \oplus \text{Auth}_{jk}$ and the data message $M_j = d_j \cdot M_i \oplus M_jQ_k$. It sends the data $\{\text{sign}_j, M_j, M, R_j, \text{AID}_j, t_5\}$ to the next level of relay device through the co-channel at the time for timestamp of t_5.

Step 4: After the next level relay device receives the request message at time t_4, it first judges whether $|t_6 - t_5| < \Delta T$. If the condition is valid, it accepts the data, otherwise it gives up the data received this time and sends a message to the data source device to request it to resend. The current relay device will compute $N_k = d_k \cdot R_j$, restore $\text{TID}_j' = AID \oplus h(N_k)$ and then compute $\text{Auth}_{jk}' = h(\text{TID}_j'||R_j||t_5||A_kQ_j)$ and $(M_j \oplus A_kQ_j) \cdot P$. The relay device (k) computes $h((M_j \oplus A_kQ_j) \cdot P)$ and compares it with $(sign_j \oplus \text{Auth}_{jk}')$, if it is equal then the authentication is successful. Then the device will verify the authenticity of the received data after calculating the data to be sent, it will calculate $(M_j \oplus A_kQ_j) \cdot P$ and send it to the higher -level data transmission device of the received data device to verify the data. The higher level device uses its own generated data $M_i \cdot Q_j$ to compare with $(M_j \oplus A_kQ_j) \cdot P$, if it is not equal it means that the device that generated the data M_j has tampered with the data, otherwise the data is fine. The after follows the above steps sequentially until the server receives the message.

The process in detail is shown in Fig. 3.

D. Data Parsing Phase

When the server receives the data after n times of relay node transmission, $\text{id}' = h(s \cdot R) \oplus MID$, then find the corresponding HID by id′.

If $\text{MAC}' = h(\text{id}'||\text{HID}||t_1||R)$ compared withMAC of authentication code, is equal, then the authentication succeeds, otherwise it fails.

Then the server will calculate $\text{key}' = h(HID||t_1||R)$ and decrypt the data $\{\text{data}\}_{\text{key}}$ to get data.

The specific process is shown in Fig. 4.

V. Security Analysis

A. Correctness Proof

Theorem 1, we will prove the correctness of how authentication is performed between devices and how the server decrypts data in the AIAS-oceanMT protocol.

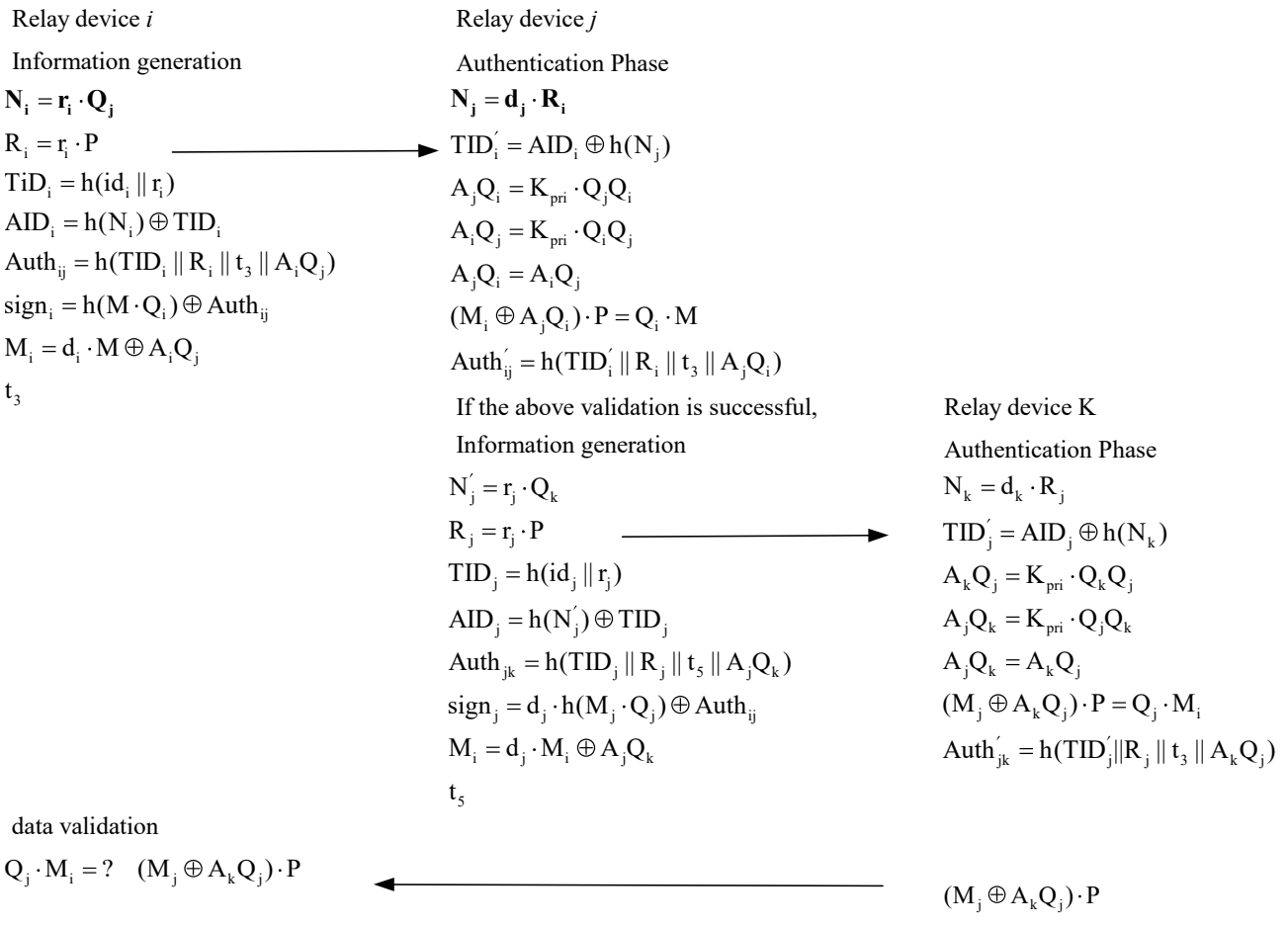


Fig. 3. Authentication process.

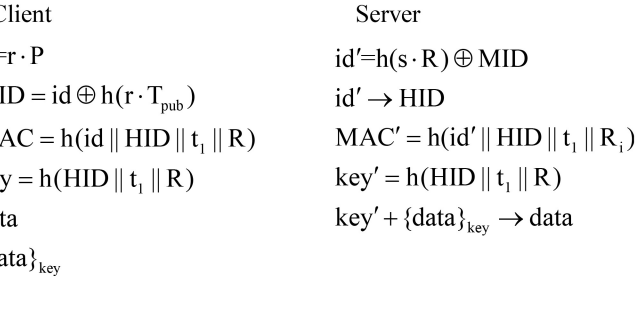


Fig. 4. Data decryption process.

Theorem 1: In step 3, relay device j receives information $\{sign_i, M_i, M, R_i, AID_i, t_3\}$. First, the relay device j determines whether the transmission time is within the specified range. Second, the relay device j calculates $N_j = d_j \cdot R_i$, followed with calculating $TID_i' = AID_i \oplus h(N_j)$. After that, the relay device uses A_j and Q_i to calculate to obtain A_jQ_i. With the data obtained from the above operations, $Auth_{ij}' = h(TID_i'||R_i||t_3||A_jQ_i)$ can be calculated. At this point, we can verify whether $h((M_i \oplus A_jQ_i) \cdot P)$ and $sign_i \oplus Auth_{ij}'$ are equal to each other. Because $N_i = r_i \cdot Q_j = d_j \cdot R_i = N_j$, $TID_i' = AID_i \oplus h(N_j) = h(N_i) \oplus TID_i \oplus h(N_j) = TID_i$. Because $A_iQ_j = K_{pri} \cdot Q_iQ_j = K_{pri} \cdot Q_jQ_i = A_jQ_i$, $Auth_{ij}' = Auth_{ij}$ and $sign_i \oplus Auth_{ij}' = h(M \cdot Q_i) \oplus Auth_{ij} \oplus Auth_{ij}' = h(M \cdot Q_i)$. Also, we could infer that $h((M_i \oplus A_jQ_i) \cdot P) = h((d_i \cdot M \oplus A_iQ_j \oplus A_jQ_i) \cdot P) = h((d_i \cdot M) \cdot P) = h(M \cdot Q_i)$. Finally, we can prove that $h((M_i \oplus A_jQ_i) \cdot P) = sign_i \oplus Auth_{ij}'$.

During the data decryption phase, the server receives the message M, calculating $s \cdot R = s \cdot r \cdot P = r \cdot T_{pub}$ and $h(s \cdot R) = h(r \cdot T_{pub})$. Because $id' = h(s \cdot R) \oplus MID = h(s \cdot R) \oplus id \oplus h(r \cdot T_{pub}) = id$ and $MAC' = h(id'||HID||t_1||R) = h(id||HID||t_1||R) = MAC$, the authentication is successful when and only when the equation holds. While the authentication is successful, you could decrypt the number$\{data\}_{key}$ to get data with a key of $key' = h(HID||t_1||R) = key$.

B. ROR Model-Based Formal Security Analysis

To verify the security of the proposed scheme, we used a real-or-random (ROR) model for formal security analysis. The details of ROR model and random oracles are as follows:

Participants: The participants associated with the protocol include all entities throughout the protocol and each of them holds a specific public and private key pair.

Partnering: Assuming that RD_i and RD_j have generated a session with $Auth_{ij}$and the identity of $ID_i$and ID_j. If RD_i and RD_j have similar sessions in the acceptable state linked directly to each other and the session have been completed successfully, they will establish a partnership with one of them acting as the initiator and the other acting as the responser.

Freshness: If an adversary issues Reveal(RD), Execute and Corrupt queries, it or its matching sessions are at risk. These queries need to be requested before the key expires. Freshness is used to distinguish fresh session keys from random session keys.

An attacker can request the following queries to violate the semantic security of the proposed protocol.

Send(RD_j, RD_k, M_2)*:* When this query is made, the adversary will send a message $M_2 = \{\mathrm{sign}_j, M_j, M, R_j, \mathrm{AID}_j, t_5\}$ to RD_k instead of RD_j. RD_k checks the validity of the query and computes the authentication information $\mathrm{Auth}'_{jk} = h(\mathrm{TID}'_j||R_j||t_5||A_kQ_j)$ referring to the above process. And the message $(M_j \oplus A_kQ_j)\cdot P$ will be returned.

Send(RD_k, RD_i)*:* The adversary sends a message $\{(M_j \oplus A_kQ_j)\cdot P\}$ to the RD_i in order to forge RD_k. When the RD_i receives the message, it checks the value of the query and computes the data validation information $M_i \cdot Q_j$ and validates the data as mentioned above. If the condition is not satisfied or the session expired, the query will be rejected.

Corrupt RD(RD)*:* The adversary can obtain the device's identity and private critical information by running this query.

Execute (*RD*)*:* It helps the adversary to obtain all the information passed in the communication when the device authenticated.

Reveal (RD)*:* It simulates an attacker with leaked session keys. If a key is generated, the device will return it as a response to the query. Otherwise, it returns null.

Test: An attacker would need to implement Reveal(RD) successfully to run this query in order to obtain the key and violate semantic security. After receiving the query, Dev will return null if no keys were generated. Otherwise, it tosses a neutral coin. If the opponent's guess (c') and the tossed coin (c) are equal, it provides the session key to the opponent. Otherwise, it generates a random value of similar length and returns it as the response.

Definition 1: Assume $\mathrm{Adv}_A^{\mathrm{AIAS-oceanMT}}(t_p)$ as the advantage of "adversary A running in polynomial time t_p and destroys the semantic security of the proposed protocol (AIAS-oceanMT) so that it derive an advantage in terms of authentication information (sign) between relay devices." Then $\mathrm{Adv}_A^{\mathrm{AIAS-oceanMT}}(t_p) = |2\Pr[c' = c] - 1|$, where c and c' denote the correct and guessed bits, respectively.

In addition, the one-way collision-resistant hash function and the Elliptic curve decision Diffie–Hellman problem (ECDDHP) are, respectively, defined in Definitions 2 and 3 to analyze the security of the proposed AIAS-oceanMT.

Definition 2: A deterministic function, such as $h : \{0, 1\}^* \rightarrow \{0, 1\}^{l_b}$, is a one-way anti-collision hash function, if it produces a fixed length l_b with any length of input string $x \in \{0, 1\}^*$ and the output string $H(x) \in \{0, 1\}^{l_b}$ of b as a hash value or message digest. Suppose the adversary A wants to find a hash collision. Then, the advantage for A to attack the hash collision is provided by$\mathrm{Adv}_A^{\mathrm{Hash}}(t_h) = \Pr[(x_1, x_2) \leftarrow rA : x_1 \neq x_2, h(x_1) = h(x_2)]$ is provided, where $Pr(X)$ denotes the probability of random event X, and $(x_1, x_2) \leftarrow rA$ denotes that the pairing is randomly chosen by A. $\mathrm{A}(\eta, t)$ shows a situation that adversary A attacks the collision resistance of $h(\cdot)$ when $\mathrm{Adv}_A^{\mathrm{Hash}}(t_h) \leq \eta$, the running time of A is at most t_h.

Definition 3: Given an elliptic curve $E_q(u, v)$ on an elliptic curve point P, ECDDHP prescribes: for a quadruple $< P, l_1.P, l_2.P, l_3.P >$, to determine whether $l_3 = l_1l_2$ or is still a unity value where $l_1, l_2, l_3 \in Z_q^*(= \{1, 2, 3, \ldots, q-1\})$.

In Theorem 2, we will prove the semantic security of AIAS-oceanMT.

Theorem 2: Suppose adversary A runs against our scheme (AIAS-oceanMT) in polynomial time t_p. q_h, $|\mathrm{Hash}|$, $\mathrm{Adv}_A^{\mathrm{ECDDHP}}(t_p)$ denote the number of hash queries, range space of one-way hash functions $h(\cdot)$ and adversary A destroys the advantage of ECDDHP at time t_p (see Definition 3), then

$$\mathrm{Adv}_A^{\mathrm{AIAS-oceanMT}}(t_p) \leq \frac{q_h^2}{|\mathrm{Hash}|} + 2\mathrm{Adv}_A^{\mathrm{ECDDHP}}(t_p). \quad (1)$$

Proof: In this article, we prove the theorem similarly to other authentication protocols [23]. We have four games, say $\mathrm{Gam}_j(j = 0, 1, 2, 3)$, which are associated with the starting and ending games Gam_0 and Gam_3, respectively. We define $\mathrm{Succ}_A^{\mathrm{Gam}_j}$ as the event that adversary A can correctly guess a random bit c in-game Gam_j and the advantage of adversary A in winning game Gam_j is $\mathrm{Adv}_{A,\mathrm{Gam}_j}^{\mathrm{AIAS-oceanMT}} = Pr[\mathrm{Succ}_A^{\mathrm{Gam}_j}]$. A detailed discussion of these games is followed.

Gam_0*:* Typically, starting the gameGam_0 is the same as the actual protocol executed under the ROR model. According to the semantic security of AIAS-oceanMT as defined in Definition 1, there are

$$\mathrm{Adv}_A^{\mathrm{AIAS-oceanMT}}(t_p) = \left|2 * \mathrm{Adv}_{A,\mathrm{Gam}_0}^{\mathrm{AIAS-oceanMT}} - 1\right|. \quad (2)$$

Gam_1*:* The Eavesdropping Attack has been modeled in this game, where adversary A can intercept all the communication messages $M = \{R, MID, MAC, t_1, \{\mathrm{data}\}_{\mathrm{key}}\}$ and $\mathrm{MSG}_i = \{R_i, \mathrm{AID}_i, \mathrm{sign}_i, M_i, t_i\}$. Finally, A can execute query Reveal and Test at the same time to confirm whether the authentication information (sign) between relay devices is a fixed number or a random number or not. The computed authentication information $\mathrm{Auth}_{ij} = h(\mathrm{TID}_i||R_i||t_3||A_iQ_j) = h(\mathrm{TID}'_i||R_i||t_3||A_jQ_i) = Auth'_{ij}$. The computed authentication information $\mathrm{h}((M_i \oplus A_jQ_i) \cdot P) = (\mathrm{sign}_i \oplus \mathrm{Auth}'_{ij})$. It is worth noting that the security of the identity message depends on the random number r_i, and the system private key k_{pri}, while the security of the authentication message depends on the security of the identity message and the private key information of the device d_i, which can not be known by eavesdropping the message M and MSG_i. Therefore, this eavesdropping attack does not increase the probability that adversary A wins in game Gam_1. As a result, both game Gam_0 and Gam_1 become indistinguishable, and we get the following result:

$$\mathrm{Adv}_{A,\mathrm{Gam}_1}^{\mathrm{AIAS-oceanMT}} = \mathrm{Adv}_{A,\mathrm{Gam}_0}^{\mathrm{AIAS-oceanMT}}. \quad (3)$$

Gam_2*:* This game contains the simulation of a hash query, R, t_1, MID and MAC in message M are randomized. Similarly, R_i, AID_i, sign_i, M_i, t_i are randomized in the message MSG_i (i is the number of the relay device). This is because the involvement of random numbers and timestamps. Therefore, collision can not occur when adversary A performs a hash

query. Since Gam_1 and Gam_2 are indistinguishable except for the simulation of the hash query contained in Gam_2. As a result of the birthday paradox, we have

$$\left|\text{Adv}_{A,\text{Gam}_2}^{\text{AIAS}-\text{oceanMT}} - \text{Adv}_{A,\text{Gam}_1}^{\text{AIAS}-\text{oceanMT}}\right| \leq \frac{q_h^2}{2|\text{Hash}|}. \quad (4)$$

Gam_3*:* In this final game, Corrupt queries have been implemented. Therefore, the attacker A can obtain the private key information d_i, identity id_i, and credential information A_i based on the execution of such queries from the corrupted device. In addition, A will have all the intercepted messages $M = \{R, MID, MAC, t_1, \{\text{data}\}_{\text{key}}\}$ and $\text{MSG}_i = \{R_i, \text{AID}_i, \text{sign}_i, M_i, t_i\}$. In order to get the original information data, adversary A needs to calculate $key = h(\text{HID}||t_1||R)$. To get the key, the adversary must know the HID. As is known, the HID contains the secret information assigned by the server and the system's private key, so the adversary can not get the data. From another point of view, the adversary destroys the transmitted data during transmission and we know that the adversary wants to tamper with the information so it is needed for the adversary to know the method to change the authentication information but this is difficult due to the ECDDHP's intractability in polynomial bounded time, which is computationally expensive. Since games Gam_2 and Gam_3 are indistinguishable from each other except for the inclusion of the Corrupt query and the ECDDHP, the advantages would follow the equation of:

$$\left|\text{Adv}_{A,\text{Gam}_3}^{\text{AIAS}-\text{oceanMT}} - \text{Adv}_{A,\text{Gam}_2}^{\text{AIAS}-\text{oceanMT}}\right| \leq \text{Adv}_A^{\text{AIAS}-\text{oceanMT}}(t_p). \quad (5)$$

Now, all the queries related to the above game have been executed. Once the Reveal query and the Test query have been executed, all that remains is to guess the random bit c. Thus, we have

$$\text{Adv}_{A,\text{Gam}_3}^{\text{AIAS}-\text{oceanMT}} = \frac{1}{2}. \quad (6)$$

From (2), (3), and (6) it can be deduced that

$$\begin{aligned}\frac{1}{2} * \text{Adv}_A^{\text{AIAS}-\text{oceanMT}}(t_p) &= \left|\text{Adv}_{A,\text{Gam}_0}^{\text{AIAS}-\text{oceanMT}} - \frac{1}{2}\right| \\ &= \left|\text{Adv}_{A,\text{Gam}_1}^{\text{AIAS}-\text{oceanMT}} - \text{Adv}_{A,\text{Gam}_3}^{\text{AIAS}-\text{oceanMT}}\right| \\ &\leq \left|Adv_{A,\text{Gam}_1}^{\text{AIAS}-\text{oceanMT}} - \text{Adv}_{A,\text{Gam}_2}^{\text{AIAS}-\text{oceanMT}}\right| \\ &\quad + \left|\text{Adv}_{A,\text{Gam}_2}^{\text{AIAS}-\text{oceanMT}} - \text{Adv}_{A,\text{Gam}_3}^{\text{AIAS}-\text{oceanMT}}\right|. \quad (7)\end{aligned}$$

From (4), (5), and (7)

$$\begin{aligned}&\frac{1}{2} * Adv_A^{\text{AIAS}-\text{oceanMT}}(t_p) \\ &\leq \frac{q_h^2}{2|\text{Hash}|} + \text{Adv}_A^{\text{ECDDHP}}(t_p). \quad (8)\end{aligned}$$

Finally, multiplying both sides of (8) by 2 at the same time, we get the desired result

$$\begin{aligned}&\text{Adv}_A^{\text{AIAS}-\text{oceanMT}}(t_p) \\ &\leq \frac{q_h^2}{|\text{Hash}|} + 2 * \text{Adv}_A^{\text{ECDDHP}}(t_p). \quad (9)\end{aligned}$$

■

C. Scyther Tool

We have implemented the proposed scheme using the Scyther Tool security authentication tool and demonstrated its resistance against attacks. It is demonstrated that the proposed protocol fulfills security claims, such as "Alive" for ensuring aliveness, "Nisynch" for noninjective synchronization, "Niagree" for noninjective agreement, "weakagree" for minimum agreement, and "secret" for confidentiality, respectively. The Scyther code and its corresponding evaluation results are presented, respectively, in Figs. 5 and 6.

D. Informal Security Analysis

Anonymity: No one can know the true identity of other devices. As mentioned earlier, devices communicate with hash values of their true identities rather than true identities. When authentication is required between devices, devices use their own private key and other information to generate temporary identity information and sent it to the next-level device. The next-level device restores the data by using its private key and the received information and then calculates the relevant information to verify whether the identity of the information source device is legitimate. During this process, no one would reveal their accurate identity information. Therefore, even if an opponent obtains relevant authentication request messages, they can not obtain accurate identity information from this information.

One-Way Authentication: Through analysis, we know that only legitimate devices can generate legitimate request messages (AID) and calculate authentication information (Auth). Therefore, devices can verify their identity information to determine whether the other party is legitimate or not.

Forward Secrecy: As seen from the protocol, internal information and index each device are unique. Additionally, a randomly chosen parameter r, and a unique timestamp t are used to establish the session key during authentication. Even if the corresponding session key is briefly leaked, it will not affect the confidentiality of communication data between other sessions.

Replay Attack: Consider capturing all messages sent between participants during the identity authentication and data authentication process on a public channel. Now, adversary may reuse this information to obtain more valuable data. To mitigate this, the system should be synchronized in time with each message containing a current timestamp. The replay attack is limited because each replayed messages includes the timestamps and nonces chosen by participants. Once an enemy attempts to act replay attack, the scheme detects the replay of old messages by verifying the relevant time information so that the system could identify replayed messages and notice the adversaries attempting to impersonate participants. And the communications would be disrupted.

Man-in-the-Middle Attack: Considering an adversary capturing all messages sent between participants during the authentication and data authentication process on a public channel, the adversary can now modify the transmitted message to make the participant believe that the received message originated from a legitimate participant. If the adversary

```
usertype TimeStamp;
const P;secret idi,idj,di,dj ,Kpri,ri,rj;hashfunction h1;secret XOR: Function;secret ScalarMulti: Function;
macro Kpub = ScalarMulti(kpri, P); macro Qi=ScalarMulti(di, P); macro Qj=ScalarMulti(dj, P);macro TIDi=h1(idi,ri);macro TIDj=h1(dj,rj);
protocol AlAS-oceanMT (Relayi, Relayj){

role Relayi {
fresh t1: TimeStamp;
macro TIDi=h1(idi,ri); macro Ri=ScalarMulti(ri,P); macro Ni= ScalarMulti(ri,Qj); macro AlDi= XOR(h1(Ni),TIDi); macro AiQj= ScalarMulti(Kpri,Qi ,Qj);
macro Authi =h1(TIDi, Ri, AiQj, t1);

send_1(Relayi, Relayj,(AlDi,Authi,Ni ,t1));
var AlDj, Authj, Rj, t2;

recv_2( Relayj ,Relayi ,(AlDj,Authj,Rj,t2));
macro Betajhat = ScalarMulti(di,Rj);macro TIDjhat= XOR(h1(Betajhat),AlDj); macro AjQi=ScalarMulti(Kpri,Qj,Qi); macro Authjhat= h1(TIDjhat, Ri, AjQi ,t2);
match (Authjhat, Authj);

claim (Relayi,Secret,idi);claim (Relayi,Secret,idj);claim (Relayi, Alive);claim (Relayi, Nisynch);claim (Relayi, Niagree);claim (Relayi, Weakagree);
claim (Relayi, Commit, Relayj , ri,rj);
};

role Relayj {

var t1;recv_1(Relayi, Relayj,(AlDi,Authi, Ni,t1));

macro Alphaihat= ScalarMulti(dj,Ni); macro TIDihat=XOR(Alphaihat,AlDi); macro AiQi=scalarMulti(Kpri,Qj,Qi); match(AjQi,AiQj);macro Authihat=h1(TIDihat,Ni,AjQi,t1);
match (Authihat,Authi);

macro TIDj1 =h1(idj, rj);macro Betaj=ScalarMulti(rj,Qi); macro AlDj= XOR(h1(Betaj), TIDj1);macro Nj=ScalarMulti(rj,P);

fresh t2: TimeStamp;
macro Rj=ScalarMulti (rj,P);macro Authj=h1(TIDj1, Rj, AjQi, t2);

send_2 (Relayj ,Relayi ,(AlDj,Authj,Nj,t2));

claim (Relayj,Secret,idi);claim (Relayj,Secret,idj);claim (Relayj,Alive);claim (Relayj,Nisynch);claim (Relayj,Niagree);claim (Relayj,Weakagree);
claim (Relayj,Commit,Relayi,ri,rj);
};
};
```

Fig. 5. Scyther code of the proposed protocol.

attempts to modify the signature, they would need to alter both d and R, which requires knowledge of the random secret $r \in Z_q^*$ and the device's private key. To modify M, the adversary would need access to the system's private key information. So that adversary cannot tamper with these messages without specific knowledge of the device. Furthermore, attempts of such an attack become impossible due to the utilization of random numbers and the current timestamp, So the protocol is resistant to "man-in-the-middle" attacks.

Impersonation Attack: To successfully execute this attack and spoof the relay device, an attacker would need to know Auth_{ij}, Auth_{jk} and $\{(M_j \oplus A_k Q_j) \cdot P\}$ to be authenticated by the protocol. However, the private key information d and identity information id of each device are inaccessible through external means, attackers are precluded from generating valid information to forge authorized devices within the system. This indicates that our proposed protocol is resistant to impersonation attacks.

Privilege-Insider Attack: Devices not internally authorized do not have access to device id and authentication information in the network. Message $\text{AID}_i = h(N_i) \oplus \text{TID}_i$ is sent during the authentication process. But only device RD_j can only obtain TID_i($N_i = r_i \cdot Q_j = d_j \cdot R_i = N_j$).Thus, unauthorized devices are prohibited from accessing other device id and authentication information.

VI. Comparative Analysis

In this section, we discuss the performance of the proposed protocols in detail and analyze our scheme in comparison with schemes [22], [23], [24]. Due to limited experimental conditions, we have used secp192r1, secp256r1, and secp512r1 to test the scheme in terms of computation overhead and communication overhead. The information of other parameters in the experiment is listed the following tables. Given the limited experimental conditions, the whole experiment is simulated on a PC with Intel i7-7700HQ @ 2.80-GHz CPU, 8-Gb RAM, and 400-GB storage. The main implementation of the experiment is done in Java, which provides rich APIs to realize various algorithmic processes in the protocol. The hash functions in this article are instantiated using SHA-256, and the elliptic curve point multiplication and point addition operations are implemented using various mathematical formula APIs provided by Java.

A. Computation Cost Comparison

When calculating the computational overhead of each operation, we use T_{ecm}, T_{eca}, and T_h to represent the "elliptic curve dot multiplication," "elliptic curve dot addition" and "||" one-way hash operation, respectively. In the proposed scheme, the time required for the bitwise XOR operation is negligible, and the time requirement of the "||" operation is also negligible. The detailed experimental data are defined in Table III.

The specific computational overhead is calculated in Table IV.

In our scheme, the data senders compute five elliptic curve dot multiplication operations and four hash operations. On the other hand, the data validators compute three elliptic curve dot multiplication operations and three hash operations. The

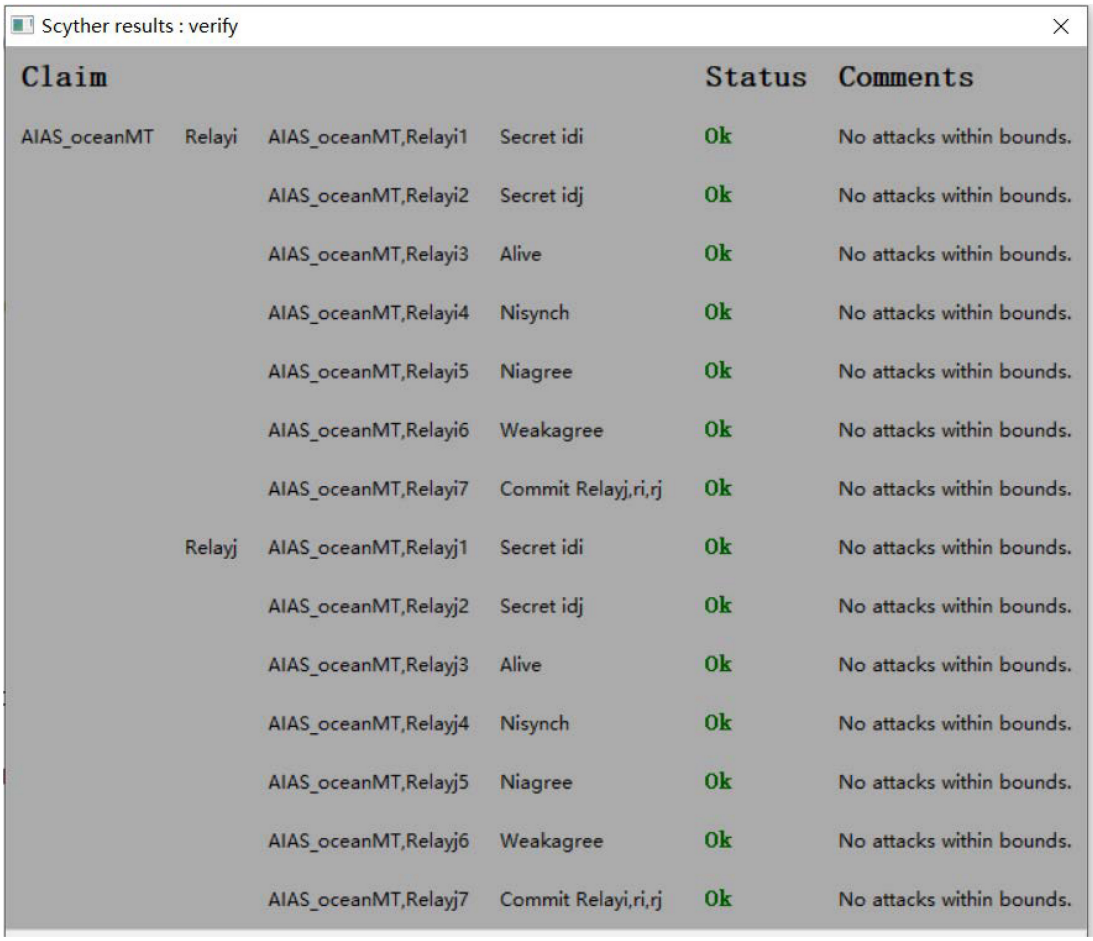


Fig. 6. Result of security analysis on the proposed protocol using Scyther tool.

TABLE III
OPERATION TIME CONSUMPTION

OPERATION	T_{ecm}	T_{eca}	T_h
secp192r1	1.54ms	0.07ms	0.56ms
secp256r1	1.83ms	0.08ms	0.56ms
secp512r1	5.13ms	0.12ms	0.56ms

total computational cost for both parties can be inferred as $8T_{\text{ecm}} + 7T_h$.

For different conditions, the computational overheads are as follows:

For Condition 1: $8 \times 1.54 + 7 \times 0.56 = 16.24$ ms.

For Condition 2: $8 \times 1.83 + 7 \times 0.56 = 18.56$ ms.

For Condition 3: $8 \times 5.13 + 7 \times 0.56 = 44.96$ ms.

In Hajian et al. [22], the data senders compute four elliptic curve dot multiplication operations and seven hash operations. And the data validators compute four elliptic curve dot multiplication operations and seven hash operations. The total computational cost for both parties is $8T_{\text{ecm}} + 14T_h$.

For different conditions, the computational overheads are as follows:

For Condition 1: $8 \times 1.54 + 14 \times 0.56 = 20.16$ ms.

For Condition 2: $8 \times 1.83 + 14 \times 0.56 = 22.48$ ms.

For Condition 3: $8 \times 5.13 + 14 \times 0.56 = 48.88$ ms.

In Srinivas et al. [23], the data senders compute three elliptic curve dot multiplication operations, seven hash operations and one elliptic curve dot addition operation. While the data validators compute three elliptic curve dot multiplication operations, seven hash operations and one elliptic curve dot addition operation. The total computational cost for both parties comes to $6T_{\text{ecm}} + 14T_h + 2T_{\text{eca}}$.

For different conditions, the computational overheads are as follows:

For Condition 1: $6 \times 1.54 + 14 \times 0.56 + 2 \times 0.07 = 17.22$ ms.

For Condition 2: $6 \times 1.83 + 14 \times 0.56 + 2 \times 0.08 = 18.92$ ms.

For Condition 3: $6 \times 5.13 + 14 \times 0.56 + 2 \times 0.12 = 38.86$ ms.

In Abbasinezhad-Mood and Nikooghadam [24], the data senders compute four elliptic curve dot multiplication operations, five hash operations and one elliptic curve dot addition operation with data validators compute four elliptic curve dot multiplication operations, five hash operations and one elliptic curve dot addition operation. The total computational cost for both parties reaches a value of $8T_{\text{ecm}} + 10T_h + 2T_{\text{eca}}$.

For different conditions, the computational overheads are as follows:

TABLE IV
COMPARISON OF COMPUTATION COSTS DURING AUTHENTICATION PHASE

	Message Sender	Data Validator	Total Time
Ours	$5T_{ecm}+4T_h$	$3T_{ecm}+3T_h$	$8T_{ecm}+7T_h$
[22]	$4T_{ecm}+7T_h$	$4T_{ecm}+7T_h$	$8T_{ecm}+14T_h$
[23]	$3T_{ecm}+7T_h+1T_{eca}$	$3T_{ecm}+7T_h+1T_{eca}$	$6T_{ecm}+14T_h+2T_{eca}$
[24]	$4T_{ecm}+5T_h+1T_{eca}$	$4T_{ecm}+5T_h+1T_{eca}$	$8T_{ecm}+10T_h+2T_{eca}$
[25]	$3T_{ecm}+6T_h+1T_{eca}$	$4T_{ecm}+4T_h+1T_{eca}$	$7T_{ecm}+10T_h+2T_{eca}$
[31]	$4T_{ecm}+4T_h+1T_{eca}$	$4T_{ecm}+4T_h+1T_{eca}$	$8T_{ecm}+8T_h+2T_{eca}$

For Condition 1: $8 \times 1.54 + 10 \times 0.56 + 2 \times 0.07 = 18.06$ ms.

For Condition 2: $8 \times 1.83 + 10 \times 0.56 + 2 \times 0.08 = 20.4$ ms.

For Condition 3: $8 \times 5.13 + 10 \times 0.56 + 2 \times 0.12 = 46.88$ ms.

In Chen et al. [25], the data senders compute three elliptic curve dot multiplication operations, six hash operations and one elliptic curve dot addition operation. The data validators compute four elliptic curve dot multiplication operations, four hash operations and one elliptic curve dot addition operation. The total computational cost for both parties is finally $7T_{\text{ecm}} + 10T_h + 2T_{\text{eca}}$.

For different conditions, the computational overheads are as follows:

For Condition 1: $7 \times 1.54 + 10 \times 0.56 + 2 \times 0.07 = 16.52$ ms.

For Condition 2: $7 \times 1.83 + 10 \times 0.56 + 2 \times 0.08 = 19.47$ ms.

For Condition 3: $7 \times 5.13 + 10 \times 0.56 + 2 \times 0.12 = 41.75$ ms.

In Garg et al. [31], the data senders compute four elliptic curve dot multiplication operations, four hash operations and one elliptic curve dot addition operation. The data validators computes four elliptic curve dot multiplication operations, four hash operations and one elliptic curve dot addition operation. The total computational cost for both parties can be calculated as $8T_{\text{ecm}} + 8T_h + 2T_{\text{eca}}$.

For different conditions, the computational overheads are as follows:

For Condition 1: $8 \times 1.54 + 8 \times 0.56 + 2 \times 0.07 = 16.94$ ms.

For Condition 2: $8 \times 1.83 + 8 \times 0.56 + 2 \times 0.08 = 19.28$ ms.

For Condition 3: $8 \times 5.13 + 8 \times 0.56 + 2 \times 0.12 = 45.76$ ms.

As can see from the computational overhead comparison Fig. 7, our scheme has a smaller computational overhead compared to other schemes, which is due to the initial design in our scheme. We use more XOR operations and hash operations to replace the more complex ECC dot addition and ECC dot multiplication operations. And in our authentication process, we use one-way authentication combined with data authentication to ensure the security of the data in the entire transmission process as well as the traceability of the transmission process. Through the above methods, our protocol has lower computational overhead, which makes it more suitable for the authentication of resource-constrained devices.

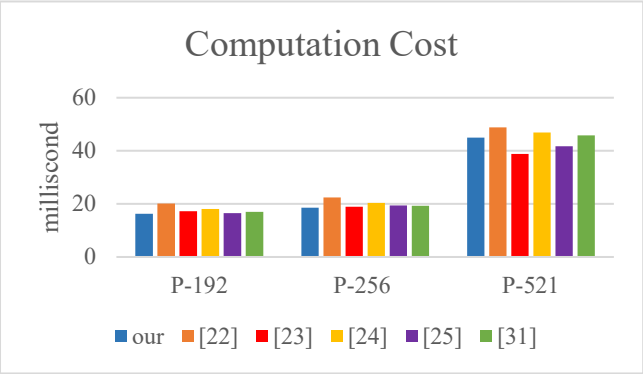


Fig. 7. Computational cost of different protocols with multi-ECC parameters.

TABLE V
COMPARISON OF COMMUNICATION COSTS DURING AUTHENTICATION PHASE

	Communication Costs
Ours	1|ECC|+2|H|+1|T|
[22]	2|ECC|+4|H|+2|T|
[23]	2|ECC|+1|H|+2|T|
[24]	2|ECC|+2|H|
[25]	4|ECC|+6|H|
[31]	2|ECC|+6|H|+4|T|

B. Communication Cost Comparison

In the communication overhead calculation of the scheme, we test the scheme based on SECP128R1, SECP160R1 and SECP192R1, where |H| represents the bits occupied after one-way hash operation, and its value is constant at 256 bits, and |T| represents the bits occupied by "timestamps," and its value is constant at 64 bits. The value of |T| represents the bits occupied by the "timestamp," which is constant at 64 bits. Details of communications overhead are shown in Table V.

In our scheme, the communication overhead required to implement the correlation algorithm for different conditions are $1 \times 192 + 2 \times 256 + 1 \times 647 = 768$ bits, $1 \times 256 + 2 \times 256 + 1 \times 64 = 832$ bits, and $1 \times 521 + 2 \times 256 + 1 \times 64 = 1097$ bits, respectively.

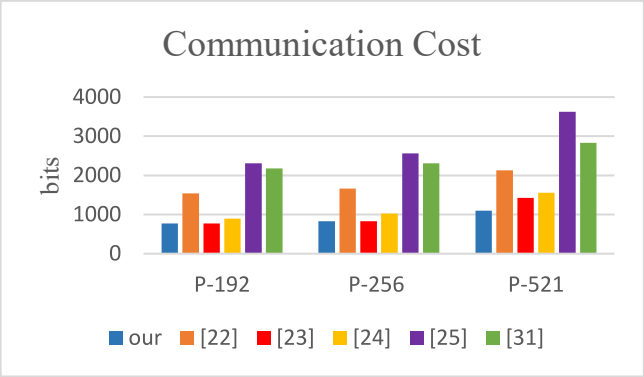


Fig. 8. Communication cost of different protocols with multi-ECC parameters.

In Hajian et al. [22], the communication overhead required to implement the related algorithms for different conditions are $2 \times 192 + 4 \times 256 + 2 \times 64 = 1536$ bits, $2 \times 256 + 4 \times 256 + 2 \times 64 = 1664$ bits, and $2 \times 521 + 4 \times 256 + 2 \times 64 = 2130$ bits, respectively.

In Srinivas et al. [23], the communication overhead required to implement the related algorithms for different conditions are $2 \times 192 + 1 \times 256 + 2 \times 64 = 768$ bits, $2 \times 256 + 1 \times 256 = 2 \times 64 = 832$ bits, and $2 \times 521 + 1 \times 256 + 2 \times 64 = 1426$ bits, respectively.

In Abbasinezhad-Mood and Nikooghadam [24], the communication overhead required to implement the related algorithms for different conditions are $2 \times 192 + 2 \times 256 = 896$ bits, $2 \times 256 + 2 \times 256 = 1024$ bits, and $2 \times 521 + 2 \times 256 = 1554$ bits, respectively.

In Chen et al. [25], the communication overhead required to implement the related algorithms for different conditions are $4 \times 192 + 6 \times 256 = 2304$ bits, $4 \times 256 + 6 \times 256 = 2560$ bits, and $4 \times 521 + 6 \times 256 = 3620$ bits, respectively.

In Garg et al. [31], the communication overhead required to implement the related algorithms for different conditions are $2 \times 192 + 6 \times 256 + 4 \times 64 = 2176$ bits, $2 \times 256 + 6 \times 256 + 4 \times 64 = 2304$ bits, and $2 \times 521 + 6 \times 256 + 4 \times 64 = 2834$ bits, respectively.

The communication overhead comparison showed in Fig. 8 reveals that our scheme has a smaller communication overhead compared to other schemes. When we design the scheme, we minimize the number of communication times in the whole process. When each relay node transmits data, it only needs to receive data from the previous relay node once without transmitting data to the previous relay node. At the same time, we perform data transmission authentication once after every relay node. In the whole authentication process, all other schemes need to communicate four times, while we only need to communicate three times to realize the authentication and data authentication.

VII. Conclusion

In this article, the security challenges associated with long-range data transmission in the complex marine network environment have been discussed and a protocol (AIAS-oceanMT) for the marine environment is introduced. Based on the (ECC) algorithm, this protocol is founded as an alternative to traditional encryption methodologies. With data transmission method through relay devices at multiple stages proposed in AIAS-oceanMT, the issue of long-distance transmission limitations could be effectively addressed. At the same time, the mechanisms for both identity and data authentication are also contained in this protocol to ensure the security and privacy of data during transmission. The security requirements is substantiated to be reached through formalized security analysis within the ROR framework. And the simulative experiments demonstrate that lower computational and communication overheads are significantly introduced with AIAS-oceanMT while ensuring security.

In the future, we plans to undertake a practical deployment in maritime settings, where a temporary platform will be established to support relay devices. And data-gathering instruments will be strategically dispersed throughout the marine environment. The AIAS-oceanMT protocol would be actually deployed and be rigorously tested under authentic and intricate oceanic network conditions. The protocol's applicability in the real-world and its adaptability to the unique challenges posed by the marine environment will be thoroughly evaluated and it will provide crucial insights into its operational viability in actual oceanic scenarios.

References

[1] R. Amin, N. Kumar, G. P. Biswas, R. Iqbal, and V. Chang, "A light weight authentication protocol for IoT-enabled devices in distributed cloud computing environment," *Future Gener. Comput. Syst.*, vol. 78, pp. 1005–1019, Jan. 2018, doi: 10.1016/j.future.2016.12.028.

[2] J. R. Wallrabenstein, "Practical and secure IoT device authentication using physical unclonable functions," in *Proc. IEEE 4th Int. Conf. Future Internet Things Cloud (FiCloud)*, Vienna, Austria, Aug. 2016, pp. 99–106, doi: 10.1109/FiCloud.2016.22.

[3] S. R. Ganta, S. P. Kasiviswanathan, and A. Smith, "Composition attacks and auxiliary information in data privacy," 2008, *arXiv:0803.0032*.

[4] M. M. Baig, J. Li, J. Liu, X. Ding, and H. Wang, "Data privacy against composition attack," in *Database Systems for Advanced Applications* (Lecture Notes in Computer Science 7238), S. Lee, Z. Peng, X. Zhou, Y.-S. Moon, R. Unland, and J. Yoo, Eds. Heidelberg, Germany: Springer, 2012, pp. 320–334. Accessed: Mar. 20, 2024. [Online]. Available: http://link.springer.com/10.1007/978-3-642-29038-1_24

[5] L. Wei, J. Cui, H. Zhong, I. Bolodurina, and L. Liu, "A lightweight and conditional privacy-preserving authenticated key agreement scheme with multi-TA model for fog-based VANETs," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 1, pp. 422–436, Jan./Feb. 2023, doi: 10.1109/TDSC.2021.3135016.

[6] H. Wang et al., "Joint biological ID: A secure and efficient lightweight biometric authentication scheme," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 3, pp. 2578–2592, May/Jun. 2023, doi: 10.1109/TDSC.2022.3186999.

[7] R. Amin, S. H. Islam, G. P. Biswas, D. Giri, M. K. Khan, and N. Kumar, "A more secure and privacy-aware anonymous user authentication scheme for distributed mobile cloud computing environments," *Security Commun. Netw.*, vol. 9, no. 17, pp. 4650–4666, Nov. 2016, doi: 10.1002/sec.1655.

[8] S. Jegadeesan et al., "An efficient anonymous mutual authentication technique for providing secure communication in mobile cloud computing for smart city applications," *Sustain. Cities Society*, vol. 49, Aug. 2019, Art. no. 101522, doi: 10.1016/j.scs.2019.101522.

[9] X. Zeng, G. Xu, X. Zheng, Y. Xiang, and W. Zhou, "E-AUA: An efficient anonymous user authentication protocol for mobile IoT," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1506–1519, Apr. 2019, doi: 10.1109/JIOT.2018.2847447.

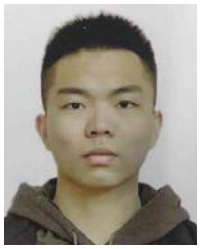
[10] L. Deng, T. Wang, S. Feng, Y. Qu, and S. Li, "Secure identity-based designated verifier anonymous aggregate signature scheme suitable for smart grids," *IEEE Internet Things J.*, vol. 10, no. 1, pp. 57–65, Jan. 2023, doi: 10.1109/JIOT.2022.3199480.
[11] L. Zhang, Y. Ye, and Y. Mu, "Multiauthority access control with anonymous authentication for personal health record," *IEEE Internet Things J.*, vol. 8, no. 1, pp. 156–167, Jan. 2021, doi: 10.1109/JIOT.2020.3000775.
[12] Y. Zheng, W. Liu, C. Gu, and C.-H. Chang, "PUF-based mutual authentication and key exchange protocol for peer-to-peer IoT applications," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 4, pp. 3299–3316, Jul./Aug. 2023, doi: 10.1109/TDSC.2022.3193570.
[13] X. Ding, X. Wang, Y. Xie, and F. Li, "A lightweight anonymous authentication protocol for resource-constrained devices in Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 1818–1829, Feb. 2022, doi: 10.1109/JIOT.2021.3088641.
[14] M. Wazid, A. K. Das, V. Odelu, N. Kumar, and W. Susilo, "Secure remote user authenticated key establishment protocol for smart home environment," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 2, pp. 391–406, Mar./Apr. 2020, doi: 10.1109/TDSC.2017.2764083.
[15] P. Gope and B. Sikdar, "Lightweight and privacy-preserving two-factor authentication scheme for IoT devices," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 580–589, Feb. 2019, doi: 10.1109/JIOT.2018.2846299.
[16] P. Kumar, A. Gurtov, M. Sain, A. Martin, and P. H. Ha, "Lightweight authentication and key agreement for smart metering in smart energy networks," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 4349–4359, Jul. 2019, doi: 10.1109/TSG.2018.2857558.
[17] B. Ying and A. Nayak, "Lightweight remote user authentication protocol for multi-server 5G networks using self-certified public key cryptography," *J. Netw. Comput. Appl.*, vol. 131, pp. 66–74, Apr. 2019, doi: 10.1016/j.jnca.2019.01.017.
[18] Q. Li, C.-F. Hsu, K.-K. R. Choo, and D. He, "A provably secure and lightweight identity-based two-party authenticated key agreement protocol for vehicular ad hoc networks," *Security Commun. Netw.*, vol. 2019, pp. 1–13, Dec. 2019, doi: 10.1155/2019/7871067.
[19] M. N. Aman, S. Taneja, B. Sikdar, K. C. Chua, and M. Alioto, "Token-based security for the Internet of Things with dynamic energy-quality tradeoff," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2843–2859, Apr. 2019, doi: 10.1109/JIOT.2018.2875472.
[20] M. A. Simplicio Jr., M. V. M. Silva, R. C. A. Alves, and T. K. C. Shibata, "Lightweight and escrow-less authenticated key agreement for the Internet of Things," *Comput. Commun.*, vol. 98, pp. 43–51, Jan. 2017, doi: 10.1016/j.comcom.2016.05.002.
[21] J. Shen, Z. Gui, S. Ji, J. Shen, H. Tan, and Y. Tang, "Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks," *J. Netw. Comput. Appl.*, vol. 106, pp. 117–123, Mar. 2018, doi: 10.1016/j.jnca.2018.01.003.
[22] R. Hajian, A. Haghighat, and S. H. Erfani, "A secure anonymous D2D mutual authentication and key agreement protocol for IoT," *Internet Things*, vol. 18, May 2022, Art. no. 100493, doi: 10.1016/j.iot.2021.100493.
[23] J. Srinivas, A. K. Das, X. Li, M. K. Khan, and M. Jo, "Designing anonymous signature-based authenticated key exchange scheme for Internet of Things-enabled smart grid systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4425–4436, Jul. 2021, doi: 10.1109/TII.2020.3011849.
[24] D. Abbasinezhad-Mood and M. Nikooghadam, "An anonymous ECC-based self-certified key distribution scheme for the smart grid," *IEEE Trans. Ind. Electron.*, vol. 65, no. 10, pp. 7996–8004, Oct. 2018, doi: 10.1109/TIE.2018.2807383.
[25] Y. Chen et al., "ECC-based authenticated key agreement protocol for industrial control system," *IEEE Internet Things J.*, vol. 10, no. 6, pp. 4688–4697, Mar. 2023, doi: 10.1109/JIOT.2022.3219233.
[26] H. Lian, Y. Yang, and Y. Zhao, "Efficient and strong symmetric password authenticated key exchange with identity privacy for IoT," *IEEE Internet Things J.*, vol. 10, no. 6, pp. 4725–4734, Mar. 2023, doi: 10.1109/JIOT.2022.3219524.
[27] S. Roy, D. Das, A. Mondal, M. H. Mahalat, B. Sen, and B. Sikdar, "PLAKE: PUF-based secure lightweight authentication and key exchange protocol for IoT," *IEEE Internet Things J.*, vol. 10, no. 10, pp. 8547–8559, May 2023, doi: 10.1109/JIOT.2022.3202265.
[28] M. N. Aman, M. H. Basheer, and B. Sikdar, "Data provenance for IoT with light weight authentication and privacy preservation," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10441–10457, Dec. 2019, doi: 10.1109/JIOT.2019.2939286.
[29] X. Yin, S. Wang, Y. Zhu, and J. Hu, "A novel length-flexible lightweight cancelable fingerprint template for privacy-preserving authentication systems in resource-constrained IoT applications," *IEEE Internet Things J.*, vol. 10, no. 1, pp. 877–892, Jan. 2023, doi: 10.1109/JIOT.2022.3204246.
[30] X. Li, C. Jiang, D. Du, M. Fei, and L. Wu, "A novel revocable lightweight authentication scheme for resource-constrained devices in cyber–physical power systems," *IEEE Internet Things J.*, vol. 10, no. 6, pp. 5280–5292, Mar. 2023, doi: 10.1109/JIOT.2022.3221943.
[31] S. Garg, K. Kaur, G. Kaddoum, J. J. P. C. Rodrigues, and M. Guizani, "Secure and lightweight authentication scheme for smart metering infrastructure in smart grid," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3548–3557, May 2020, doi: 10.1109/TII.2019.2944880.

Jun Ye received the Ph.D. degree form Xidian University, Xi'an, China, in 2017.

He is currently an Associate Professor and the Ph.D. Supervisor with the School of Cyberspaces Security, Hainan University, Haikou, China. His research interests include applied cryptography, privacy protection, and cloud computing.

Xinhui Cao received the B.E. degree in software engineering from Jiangxi Agricultural University, Nanchang, China, in 2021. He is currently pursuing the master's degree in cyberspace security with Hainan University, Haikou, Hainan, China.

His research interests include information security and network security.



Shaoxiong Xie received the B.E. degree in information security from Anhui University of Technology, Maanshan, China, in 2020. He is currently pursuing the master's degree in cyberspace security with Hainan University, Haikou, Hainan, China.

His research interests include privacy computing and network security.