Non-interactive Identity-Based Underwater Data Transmission With Anonymity and Zero Knowledge

Changsheng Wan, Vir Virander Phoha, Yuzhe Tang, and Aiqun Hu

***Abstract*—Due to the lack of security infrastructures for underwater wireless communications among vehicles, data transmission protocols typically use identity-based cryptography for protecting transmitted data. However, current identity-based cryptographic schemes transmit vehicles' real identities along with the messages, which makes the communication schemes vulnerable to attacks. For example, the adversaries can infer real identities of the vehicles and thus collect important information about underwater vehicles, track them, and thus be in an advantageous position to attack them. In addition, during data transmission, adversaries can count the number of underwater vehicles that are communicating and thus evaluate the enemy's scale of operations. At the same time, due to the complex underwater environment, transmitted messages can be easily lost. Therefore, non-interactive data transmission schemes that ensure the underwater vehicles do not transmit additional messages for authentication and key establishment are needed. To address these needs, we present two novel non-interactive identity-based data transmission protocols. Similar to the protocols in this field, our protocols provide integrity and confidentiality protections for transmitted messages. However, as opposed to the other well-known approaches, our protocols do not expose information of vehicles' real identities and use different identities for transmitting each message. By doing so, our protocols provide protection against adversaries' collection of information about underwater vehicles. Moreover, in our protocols, underwater vehicles can transmit messages directly, without previously sending authentication and key establishment messages, thus achieving the non-interactivity goal. In addition, one of our protocol even permits vehicles transmitting messages without knowing any information about each other. Therefore, we posit that our protocols are quite suitable for transmitting messages for underwater environment. Experimental results show that the proposed protocols are feasible for real-world applications.**

***Index Terms*—Anonymity, identity-based data transmission, non-interactivity, underwater wireless communications, zero-knowledge.**

Manuscript received August 12, 2017; revised September 12, 2017 and September 25, 2017; accepted September 25, 2017. Date of publication September 28, 2017; date of current version February 12, 2018. This work was supported in part by National Natural Science Foundation of China under Grants 61101088 and 71402070, in part by Opening Project of Key Lab of Information Network Security of Ministry of Public Security (C16604), in part by Natural Science Foundation of Jiangsu Province (BK20161099), and in part by National Science Foundation (Award SaTC No.: 1527795). The review of this paper was coordinated by Prof. Alfredo Luigi Grieco. *(Corresponding author: Changsheng Wan).*

C. Wan is with the School of Information Science and Engineering, Southeast University, Nanjing, Jiangsu 210096, China (e-mail: wan.changsheng@163.com).

V. V. Phoha and Y. Tang are with the Syracuse University, Syracuse, NY 13244 USA (e-mail: vvphoha@syr.edu; ytang100@syr.edu).

A. Hu is with the School of Information Science and Engineering, Southeast University, Nanjing, Jiangsu 210096, China (e-mail: aqhu@seu.edu.cn).

Digital Object Identifier 10.1109/TVT.2017.2757500

I. Introduction

OVER the past years, "Underwater Wireless Communications (UWC)" technology has been widely deployed in a variety of real-world applications, such as national security and defense, collection of scientific data from ocean-bottom stations, pollution monitoring in environmental systems, remote control in the off-shore oil industry and disaster detection and early warning [1], [2]. To ensure the security of communications between underwater vehicles, data transmission protocols have been developed for providing integrity and confidentiality protections to the transmitted messages [3].

Regardless of the technology implemented, as shown in Fig. 1, a typical data transmission protocol for UWC systems [4] includes three entities: the gateway (GW), the underwater vehicle A (V_A) and the underwater vehicle B (V_B). In practice, these three entities are involved in two phases, namely the initialization phase and the data transmission phase. During the initialization phase, the GW generates and deploys keying materials for V_A and V_B. During the data transmission phase, V_A signs and encrypts messages and sends them to V_B. In this data transmission protocol, the transmitted messages should not be tampered or decrypted by an adversary. Otherwise, the UWC system may collect incorrect data, leading to wrong decisions.

Lack-of-security-infrastructure is a serious concern for underwater data transmission protocols. Due to the complicated communication environment and limited energy of underwater vehicles [1], [2], it is not suitable to deploy traditional security infrastructures such as "Public Key Infrastructure (PKI)" [5], [6] and symmetric key distribution infrastructure [7]. This is illustrated by the following three examples. First, due to the wide area of oceans, it is expensive to deploy security infrastructures such as "Certificate Authorities (CAs)" [5], [6] in these oceans. Second, since the communication channels in underwater wireless networks can be seriously affected by lots of factors (e.g., the marine environment, noise, limited bandwidth and power resources, and the harsh underwater ambient conditions), it is not suitable for vehicle A and vehicle B to consult a third party such as CA in security infrastructures during data transmission. Third, since underwater vehicles may be low-energy sensors, it is difficult and cost prohibitive for these sensors to communicate with a third party such as CA in security infrastructures during data transmission as transmitting and receiving message will consume energy. Therefore, it is desired to use "Identity-Based Cryptography (IBC)" [8] techniques for protecting transmitted data. By using IBC, the GW generates and deploys keying materials for V_A and V_B from their identities, while V_A and V_B

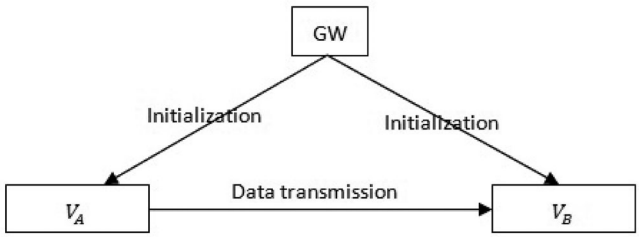


Fig. 1. Data transmission overview.

just use these keying materials for protecting transmitted data without consulting GW during data transmission. Unfortunately, current underwater protocols (i.e., [9]–[30]) are still based on traditional security infrastructures.

Anonymity is another serious concern for underwater data transmission protocols. Due to the openness of underwater wireless channel, it is easy for malicious adversaries to get the identities included in the transmitted data, track underwater vehicles, and attack these vehicles. For example, if the underwater vehicle is a military submarine, the enemy may capture it if he can track the vehicle. Moreover, if the vehicles use fixed identities, the enemy may easily count underwater vehicles. Therefore, it is desired to design a data transmission protocol with anonymity, where underwater vehicles use changeable identities instead of one fixed identity. By doing so, the attackers will be confused, and make the mistake that there are a lot of vehicles in this area. However, in current underwater security protocols, the anonymity feature has been largely neglected.

Zero-knowledge is the third serious concern for underwater data transmission protocols. In underwater wireless environment, when two vehicles meet, they may know nothing about each other and cannot transmit their real identities. Otherwise, the enemy may get their real identities and know that a submarine army is nearby, resulting in failure of a military action. In this case, vehicles must be able to transmit messages without knowing any knowledge of each other.

Non-interactivity is the fourth serious concern for underwater data transmission protocols. In underwater wireless environment, messages may be easily lost due to the poor wireless channels [1], [2]. In this case, the subsequent communication may be delayed. Therefore, when two underwater vehicles meet, they should send messages directly, without transmitting additional messages for authentication and key establishment.

In summary, designing a non-interactive identity-based data transmission protocol for underwater wireless communications is a nontrivial task due to the following five reasons. First, underwater vehicles cannot expose any information about their real identities. Second, sometimes vehicles know nothing about each other. Third, underwater wireless communications require that the enemy cannot count vehicles. Fourth, the underwater environment lacks infrastructure for achieving the above security goals. Fifth, messages are easily lost due to poor underwater wireless channels. More importantly, when focusing on this research topic, we find that there is no cryptographic primitive which can be directly applied to address all the above issues. In [31], the authors designed two secure data transmission protocols, which are identity-based and can address the non-interactivity issue. However, these two protocols still lack the above anonymity and zero-knowledge features. In Section II, we'll further analyze current security protocols for underwater communications to arrive at this conclusion. Motivated by this observation, this paper mainly makes four contributions:

1) We analyze the security requirements for underwater data transmission, and then present a comprehensive set of design goals for the protocol of this kind.

2) We design two novel non-interactive identity-based data transmission protocols for underwater wireless communications. The first protocol is based on bilinear map [32] with lower storage cost and higher computation cost and has both anonymity and zero-knowledge features, which is suitable for underwater vehicles with high computing power. And the second one is based on algebraic signature [33] with higher storage cost and lower computation cost and has only the anonymity feature, which is suitable for underwater vehicles with high memory. Our protocols are different from traditional protocols in two ways. First, in our protocols, keying materials are generated from identities of underwater vehicles. By doing so, vehicles can authenticate each other without consulting security infrastructures during data transmission. Therefore, the above Lack-of-security-infrastructure problem is solved. Second, in our protocols, vehicles use cryptographically generated one-time identity for transmitting each message. By doing so, malicious adversaries will be confused and cannot track underwater vehicles. Third, in the bilinear-map-based protocol, vehicles use randomly generated temporary identities for discovering neighbors. By doing so, vehicles can transmit messages without knowing any knowledge of each other. Fourth, in both protocols, underwater vehicles can send messages directly without previously transmitting additional messages for authentication and key establishment. By doing so, vehicles can reduce the risk of message-loss. Therefore, our protocols are quite suitable for underwater environment.

3) We analyze the security of proposed protocols in the random oracle model [34], showing it can achieve security goals.

4) We evaluate the performance of proposed protocols, showing it is feasible for real world applications.

The remainder of this paper is organized as follows. First, in Section II, we survey the related work and discuss the security and efficiency issues in current protocols. Second, in Section III, we describe the proposed protocols in detail. Third, we present security analysis and performance evaluation for proposed protocols in Section IV and Section V, respectively. Finally, we draw our conclusions in Section VI.

II. Related Work

Due to the openness of wireless channels between underwater vehicles, security is very important for data transmission in underwater wireless communications. Therefore, lots of security protocols (i.e., [9]–[30]) have been designed for this purpose. All protocols for securing transmitted data in UWC can be categorized into five types: physical layer security,

"Media Access Control (MAC)" layer security, routing layer security, data transmission layer security and application layer security.

The authors in [9] and [10] discussed the jamming attacks in physical layer. Recent work [11] used analog network coding for securing physical layer communications. The work in [12] analyzed wireless channel models in underwater environment. The authors in [13] and [14] used underwater wireless channel for generating shared keys between vehicles. The security issue for MAC layer is time synchronization, as discussed in [15]–[17]. There are a lot of attacks on routing protocols in underwater environment such as flooding attack, Sybil attack, wormhole attack, sinkhole attack, and black hole attacks [18]–[21]. The application layer security issues mainly refer to secure localization and trust management as discussed in [29], [30].

Security issues in the data transmission layer mainly refer to key management, integrity and confidentiality. Recent work [22] tested the time costs of traditional signing algorithms such as elliptic curve signing algorithm. The work in [23] is a key management scheme for underwater wireless communication. And the work in [24] is a framework of integrity and confidentiality protection. The authors in [25], [26] and [27] discussed the use of symmetric-key-based signing and encryption algorithms in underwater wireless communications.

However, all the above data transmission schemes for underwater wireless communication systems have the following problems.

1) They are built on security infrastructures for key management, which is not suitable for underwater environment as discussed in Section I.

2) They lack the non-interactivity feature as illustrated by the following examples. First, for symmetric-key-based schemes, when two underwater vehicles want to communicate with each other, they have to generate a shared key for protecting transmitted data. Since they may have no pre-established shared keys, these two underwater vehicles have to consult security infrastructure, which may be impossible in underwater environment. Second, for asymmetric-key-based schemes, these two underwater vehicles have to consult security infrastructure for verifying public keys, which may be impossible and expose real identities of vehicles.

3) They largely neglect the anonymity and zero-knowledge features, resulting in a lot of issues as discussed in Section I. Therefore, it is desired to design a non-interactive identity-based data transmission protocol with anonymity and zero-knowledge for underwater wireless communications.

Recently, several non-interactive identity-based data transmission protocols [31] have been designed, called SET-IBS and SET-IBOOS. These two protocols are initially designed for wireless sensor networks, and can potentially be deployed in underwater wireless communication systems. However, these two protocols still lack the following features which are important for underwater wireless communications.

1) They lack the anonymity feature. Since both SET-IBS and SET-IBOOS transmit real identities of vehicles, these two protocols lack the anonymity feature.

TABLE I
FEATURES OF CURRENT DATA TRANSMISSION PROTOCOLS

	[22]	[23]	[24]	[25]	[26]	[27]	SET-IBS	SET-IBOOS
Identity-based	×	×	×	×	×	×	√	√
Anonymity	×	×	×	×	×	×	×	×
Zero-knowledge	×	×	×	×	×	×	×	×
Non-interactivity	×	×	×	×	×	×	√	√
Integrity	√	×	√	√	√	√	√	√
Confidentiality	×	×	√	√	√	√	×	×

2) They lack the zero-knowledge feature. Since vehicles know real identities of each other, both SET-IBS and SET-IBOOS lack the zero-knowledge feature.

3) They lack the confidentiality feature. Both SET-IBS and SET-IBOOS only provide integrity protection, and cannot provide confidentiality protection for transmitted messages.

The features for the above data transmission protocols are listed in Table I. From Table I, it can be seen that, current data transmission protocols for underwater wireless communications have many problems, which prevent them from being deployed in underwater environment. Motivated by the above observation, it is desired to design a non-interactive identity-based underwater data transmission protocol with anonymity and zero-knowledge.

III. THE PROPOSED PROTOCOLS

In this section, we first describe the preliminaries and design goals for secure underwater data transmission protocols. Then, we give the two identity-based data transmission protocols. One is based on bilinear map and the other is based on algebraic signature.

A. Preliminaries

1) Bilinear Map: Let G and G_T be two groups with the same order q, and g is the generator of G. Then, a bilinear map group [32] is denoted by $e : G \times G \rightarrow G_T$, which fulfills the following requirements:

1) *Bilinearity:* The map e is symmetric since $\forall x, y \in Z_q$, $e(g^x, g^y) = e(g, g)^{xy} = e(g^y, g^x)$. $\forall A, B \in G$, $e(XY, g) = e(X, g)e(Y, g)$.

2) *Non-Degeneracy:* $\exists X, Y \in G, e(X, Y) \neq 1$.

3) *Computability:* For $\forall X, Y \in G$, it is efficient to compute $e(X, Y)$.

2) Algebraic signature: Given a set of secret keys $sk = \{sk_1, ..., sk_n \in Z_q\}$ and a string $str = x_1x_2...x_n$ where $x_1, x_2, ..., x_n \in \{0, 1\}$, the algebraic signature [33] for str is computed as $\sigma_{str} = \sum_{i=1}^{n} x_i sk_i \bmod q$.

B. Design Goals

A non-interactive identity-based data transmission protocol for UWC with anonymity and zero-knowledge should achieve the following design goals:

1) Anonymity: It should be guaranteed that the underwater vehicles use one-time identity when transmitting each message.

TABLE II
NOTATIONS IN THIS PAPER

Notation	Description
V_A, V_B	Real identities of the vehicles
sk_{GW}, pk_{GW}, pub	Public and private parameters of the GW
TID_A, TID_B	temporary identities of underwater vehicles
sk_A, sk_B	Privates keys of the vehicles
OTI_A, OTI_B	One-time identities of underwater vehicles
$para_A, para_B$	Parameters for the one-time identities
M, σ, τ	Transmitted message, its signature and ciphertext
sk	Generated shared key between vehicles
G, G_T, g, q	Groups, generator and prime order
h_0, h_1, h_2, h_3	Hash functions
r_A, r_B, N_A, N_B	Random numbers
PK_{N_A}, PK_{N_B}	public keys of N_A and N_B

By doing so, the adversary will not be able to track underwater vehicles or count them.

2) Zero-knowledge: It should be guaranteed that underwater vehicles who do not know any information of each other can transmit messages without exposing their real identities.

3) Confidentiality: It should be guaranteed that the adversary cannot decrypt transmitted messages in this protocol.

4) Integrity It should be guaranteed that the adversary cannot tamper transmitted messages in this protocol.

5) Non-interactivity: It should be guaranteed that vehicles do not need transmit additional messages for authentication and key establishment before data transmission.

6) High performance: It should be guaranteed that the computation, communication and storage costs are low.

C. The Two Identity-Based Data Transmission Protocols

In this subsection, we describe the "Bilinear-Map-Based Protocol (BMBP)" and "Algebraic-Signature-Based Protocol (ASBP)" in detail. The notations used in this paper are displayed in Table II. And the system model of these two protocols is shown in Fig. 2, which includes three phases, namely the initialization phase, the neighbor discovery phase, and the data transmission phase.

Note that the neighbor discovery phase is only defined for the bilinear-map-based protocol, because we assume V_A and V_B has no information of each other, and they generate temporary identities for communication during this phase. In contrast, the algebraic-signature-based protocol has no neighbor discovery phase, because we assume V_A and V_B know real identities of each other. Therefore, by adding a neighbor discovery phase, the bilinear-map-based protocol can work in difficult environment, where V_A and V_B do not know any information of each other.

1) The Initialization Phase: The initialization phase is defined for both BMBP and ASBP. During this phase, the GW initializes the cryptographic system and deploys cryptographic parameters to underwater vehicles.

First of all, for initializing the cryptographic system, the GW generates its own private key (sk_{GW}), and the set of public cryptographic parameters (pub) using the $InitSys$ algorithm defined in Algorithm 1 and Algorithm 2.

Algorithm 1: Initialization phase of BMBP.

procedure $InitSys$ **algorithm**
Output: The set of public cryptographic parameters (pub) and its own private key (sk_{GW}).
Step 1: GW creates a pairing group $e : G \times G \rightarrow G_T$ with the prime order q, and the generator $g \in G$.
Step 2: GW randomly generates its own private key $sk_{GW} \in Z_q$.
Step 3: GW computes its own public key $pk_{GW} = g^{sk_{GW}} \in G$.
Step 4: GW gets the set of public cryptographic parameters $pub = \{G, G_T, q, g, pk_{GW}\}$.
end procedure $InitSys$ **algorithm**

procedure $Gensk$ **algorithm**
Input: The vehicle's real identity ID (e.g., V_A or V_B), the set of public parameters (pub), and the GW's private key (sk_{GW}).
Output: The vehicle's private key sk_{ID} (e.g., sk_A or sk_B).
Step 1: GW computes $sk_{ID} = ID^{sk_{GW}} \in G$.
end procedure $Gensk$ **algorithm**

Then, when a vehicle wants to join the data transmission system, the GW generates a private key (sk_{ID}) for it from the vehicle's real identity (ID) and deploys (sk_{ID} and pub to the underwater vehicle over their pre-established secure channel. This key-generating algorithm ($Gensk$) is defined in Algorithm 1 and Algorithm 2.

Comparing the $InitSys$ algorithms defined in Algorithm 1 and Algorithm 2, we can see that, both BMBP and ASBP output sk_{GW} and pub, which will be used for generating keying materials for vehicles. However, in ASBP, the GW needs to perform more modular exponentiations and store more public and private keys. Therefore, in ASBP, the computation and storage costs on the GW is higher. However, since the $InitSys$ algorithm is run only once and the GW is typically a power device, these costs can be omitted, and we mainly focus on the data transmission phase when evaluating the performance of the newly designed protocols in this paper.

Comparing the $Gensk$ algorithms defined in Algorithm 1 and Algorithm 2, we can see that, the BMBP algorithm uses a modular exponentiation for generating vehicle's private key, while ASBP only uses several light-weight modular additions. So, the computation cost of ASBP is lower.

After the initialization phase, the GW holds (pub, sk_{GW}), the underwater vehicle with the real identity V_A holds (V_A, sk_A, pub), and the underwater vehicle with the real identity V_B holds (V_B, sk_B, pub). Note that, though vehicles store similar keying materials in BMBP and ASBP, the storage cost of ASBP is a little higher. This is because ASBP has to store more public keys in pub.

2) The Neighbor Discovery Phase: The neighbor discovery phase is defined only for BMBP, where V_A and V_B know nothing of each other and they have to generate temporary identities for communication. In contrast, in ASBP, we assume V_A and V_B

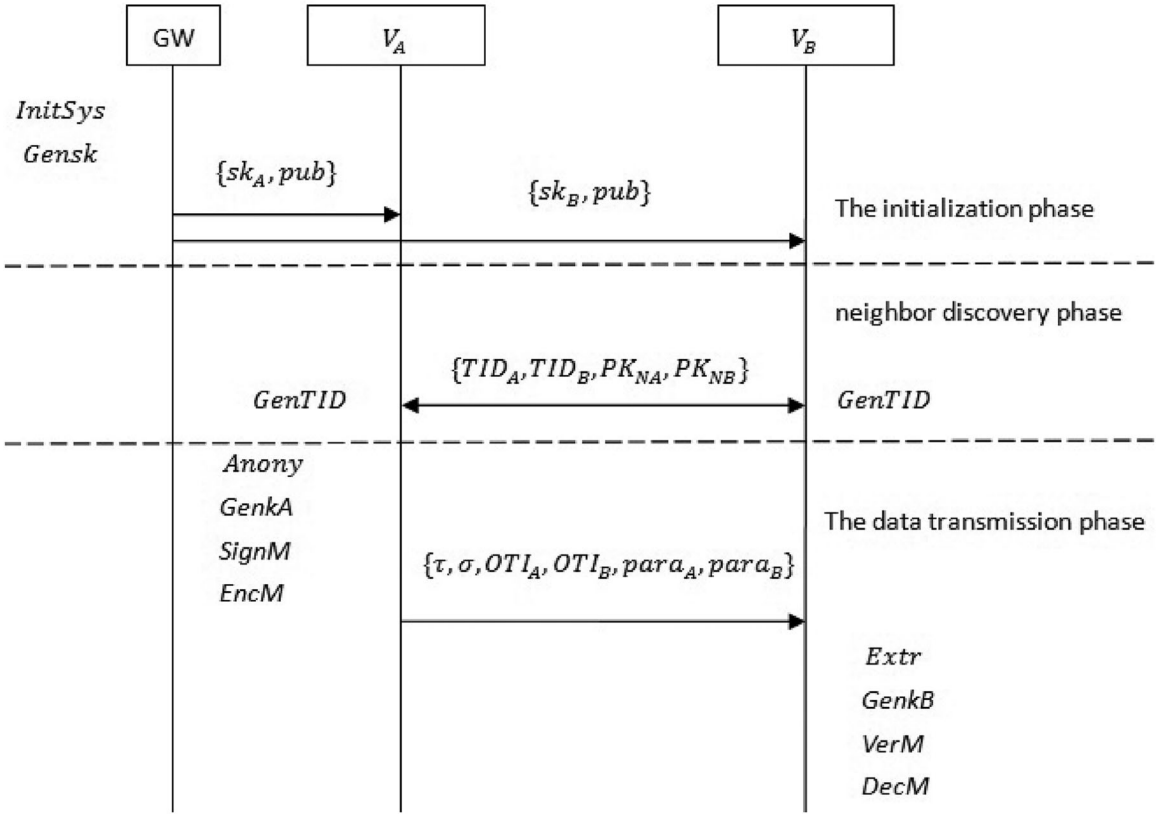


Fig. 2. System model of the two protocols.

Algorithm 2: Initialization phase of ASBP.

procedure $InitSys$ **algorithm**
Output: The set of public cryptographic parameters (pub) and its own private key (sk_{GW}).
Step 1: GW creates a group G with the prime order q, and the generator $g \in G$.
Step 2: GW randomly generates a set of private keys $\{sk_{GW1}, sk_{GW2}, \cdots, sk_{GWn} \in Z_q\}$.
Step 3: GW computes the corresponding set of public keys as $\{pk_{GW1} = g^{sk_{GW1}}, pk_{GW2} = g^{sk_{GW2}}, \cdots, pk_{GWn} = g^{sk_{GWn}}\}$.
Step 4: GW gets its own private key $sk_{GW} = \{sk_{GW1}, sk_{GW2}, \cdots, sk_{GWn} \in Z_q\}$ and the set of public parameters $pub = \{G, q, g, pk_{GW1}, pk_{GW2}, \cdots, pk_{GWn} \in G\}$. **end procedure** $InitSys$ **algorithm**

procedure $Gensk$ **algorithm**
Input: The vehicle's real identity ID (e.g., V_A or V_B), the set of public parameters (pub), and the GW's private key (sk_{GW}).
Output: The vehicle's private key sk_{ID} (e.g., sk_A or sk_B).
Step 1: GW computes $h_1(ID) = x_1 \cdots x_n$, where $h_1 : G \rightarrow \{0,1\}^n$ is a hash function, and $x_1, \cdots, x_n \in \{0,1\}$.
Step 2: GW computes $sk_{ID} = \sum_{i=1}^{n} x_i * sk_{GWi} \bmod q$.
end procedure $Gensk$ **algorithm**

Algorithm 3: Neighbor discovery phase of BMBP.

procedure $GenTID$ **algorithm**
Input: The vehicle's real identity ID (e.g., V_A or V_B), the randomly generated nonce N_{ID} (e.g., N_A or N_B), and the set of public parameters (pub).
Output: A temporary identity TID_{ID} (e.g., TID_A or TID_B) and the corresponding public key $PK_{N_{ID}}$ for the random nonce (e.g., PK_{N_A} or PK_{N_B}).
Step 1: The vehicle computes $TID_{ID} = ID^{N_{ID}} \in G$.
Step 2: The vehicle computes $PK_{N_{ID}} = g^{N_{ID}}$.
end procedure $GenTID$ **algorithm**

know real identities of each other, and the neighbor discovery phase is avoided.

In the neighbor discovery phase, when V_A and V_B meet each other in the underwater environment, each of them randomly generates a nonce $N_A \in Z_q$ (or $N_B \in Z_q$) and then generates a temporary identity from its real identity and the nonce and broadcasts it. This identity will be used for temporarily identifying the vehicle. The generating algorithm $GenTID$ is defined in Algorithm 3.

From Algorithm 3, it can be seen that, both V_A and V_B need to generate and broadcast their own temporary identity. Therefore, this phase will consume more computation and communication costs. However, this phase enables the two vehicles to transmit messages without knowing any information of each other. Therefore, BMBP can work in difficult environment.

Algorithm 4: $Anony$ algorithm for BMBP.

procedure $Anony$ **algorithm**
Input: V_A's temporary identity (TID_A), V_B's temporary identity (TID_B), and the set of public parameters (pub).
Output: V_A's one-time identity (OTI_A), V_B's one-time identity (OTI_B), the corresponding parameter for OTI_A ($para_A$), and the corresponding parameter for OTI_B ($para_B$).
Step 1: V_A generates two random numbers $r_A, r_B \in Z_q$.
Step 2: V_A computes the one-time identities $OTI_A = TID_A{}^{r_A} \in G$ and $OTI_B = g^{r_B} \in G$.
Step 3: V_A computes $t = e(TID_B, pk_{GW})^{r_B} \in G_T$.
Step 4: V_A computes the corresponding parameter for OTI_A as $para_A = r_A + h_0(t) \in Z_q$, where $h_0 : G_T \rightarrow Z_q$ is a hash function.
Step 5: V_A computes the corresponding parameter for OTI_B as $para_B = r_B + h_0(t) \in Z_q$.
end procedure $Anony$ **algorithm**

After the neighbor discovery phase, V_A gets $(TID_A, TID_B, PK_{N_A}, PK_{N_B}, N_A)$, and V_B gets $(TID_A, TID_B, PK_{N_A}, PK_{N_B}, N_B)$.

3) The Data Transmission Phase: The data transmission phase is defined for both BMBP and ASBP. This phase is run between the underwater vehicle V_A and the underwater vehicle V_B, and includes four steps as described below.

Step 1: When V_A wants to transmit a message to V_B, it first generates the one-time identities (OTI_A and OTI_B) for anonymity and the corresponding parameters ($para_A$ and $para_B$) using the following $Anony$ algorithm. A little difference between BMBP and ASBP is that, BMBP uses temporary identities (TID_A and TID_B) for generating OTI_A and OTI_B while ASBP uses V_A and V_B. This is because, BMBP assumes these two vehicles know nothing of each other while ASBP assumes these two vehicles know real identities of each other. The anonymization algorithms ($Anony$) for BMBP and ASBP are defined in Algorithm 4 and Algorithm 5, respectively.

Algorithm 4 and Algorithm 5 are different in two ways. (1) When computing OTI_A, BMBP uses TID_A while ASBP uses V_A. This is because, vehicles know real identities of each other in ASBP, while they do not in BMBP. However, the computation costs for generating OTI_A are the same. (2) When computing t, BMBP uses one pairing and one modular exponentiation, while ASBP only uses several light-weight modular multiplications. Therefore, the total computation cost of ASBP in the $Anony$ algorithm is much lower than that of BMBP.

Step 2: After generating one-time identities using the above $Anony$ algorithm, V_A generates a shared key sk using the following $GenkA$ algorithm, and then signs and encrypts the message using sk and the following $SignM$ and $EncM$ algorithms, respectively. Finally, V_A sends the encrypted message (τ), the signature (σ), the one-time identities (OTI_A and OTI_B), and the corresponding parameters ($para_A$ and $para_B$) to V_B. The message-sending processes of BMBP and ASBP are defined in Algorithm 6 and Algorithm 7, respectively.

Algorithm 5: $Anony$ algorithm for ASBP.

procedure $Anony$ **algorithm**
Input: V_A's real identity (V_A), V_B's real identity (V_B), and the set of public parameters (pub).
Output: V_A's one-time identity (OTI_A), V_B's one-time identity (OTI_B), the corresponding parameter for OTI_A ($para_A$), and the corresponding parameter for OTI_B ($para_B$).
Step 1: V_A generates two random numbers $r_A, r_B \in Z_q$.
Step 2: V_A computes the one-time identities as $OTI_A = V_A{}^{r_A}$ and $OTI_B = g^{r_B}$.
Step 3: V_A computes $h_1(V_B) = b_1 \cdots b_n$, where $h_1 : G \rightarrow \{0,1\}^n$ is a hash function, and $b_1, \cdots, b_n \in \{0,1\}$.
Step 4: V_A computes $t = \prod_{i=1}^{n} pk_{GW\,i}^{b_i} \in G$.
Step 5: V_A computes the corresponding parameter for OTI_A as $para_A = r_A + h_2(t) \in Z_q$, where $h_2 : G \rightarrow Z_q$ is a hash function.
Step 6: V_A computes the corresponding parameter for OTI_B as $para_B = r_B + h_2(t) \in Z_q$.
end procedure $Anony$ **algorithm**

Algorithm 6: Message-sending processes of BMBP.

procedure $GenkA$ **algorithm**
Input: V_A's private key (sk_A), V_A's temporary identity (TID_A), V_B's temporary identity (TID_B), V_A's nonce generated in the neighbor discovery phase (N_A), and the set of public parameters (pub).
Output: A shared key (sk).
Step 1: V_A computes $sk = e(sk_A, TID_B^{N_A}) \in G_T$.
end procedure $GenkA$ **algorithm**

procedure $SignM$ **algorithm**
Input: The shared key (sk), the message to be transmitted (M), and the set of public parameters (pub).
Output: A signature (σ) for M.
Step 1: V_A computes $\sigma = h_3(h_0(sk), M) \in Z_q$, where $h_0 : G_T \rightarrow Z_q$ and $h_3 : Z_q \rightarrow Z_q$ are hash functions.
end procedure $SignM$ **algorithm**

procedure $EncM$ **algorithm**
Input: The shared key (sk), the message to be transmitted (M), and the set of public parameters (pub).
Output: The ciphertext (τ) for M.
Step 1: V_A computes $\tau = h_0(sk) + M \bmod q$.
end procedure $EncM$ **algorithm**

Comparing $GenkA$ in Algorithm 6 and Algorithm 7, we can see that, the $GenkA$ in BMBP uses one pairing and one modular exponentiation, while that in ASBP uses only one modular exponentiation and several modular multiplications. Therefore, the total computation cost of ASBP in the $GenkA$ algorithm is much lower than that of BMBP. In addition, $sk \in G_T$ in BMBP, while $sk \in G$ in ASBP. This will result in different signing and encryption constructions of these two protocols as shown below.

Comparing $SignM$ and $EncM$ in Algorithm 6 and Algorithm 7, we can see that, the constructions of $SignM$ and

Algorithm 7: Message-sending processes of ASBP.

procedure $GenkA$ **algorithm**
Input: V_A's private key (sk_A), V_A's real identity (V_A), V_B's real identity (V_B), and the set of public parameters (pub).
Output: A shared key (sk).
Step 1: V_A computes $sk = \left(\prod_{i=1}^{n} pk_{GW\,i}^{b_i}\right)^{sk_A} \in G$, where $b_1, \cdots, b_n \in \{0,1\}$ is the same as that described in the $Anony$ algorithm.
end procedure $GenkA$ **algorithm**

procedure $SignM$ **algorithm**
Input: The shared key (sk), the message to be transmitted (M), and the set of public parameters (pub).
Output: A signature (σ) for M.
Step 1: V_A computes $\sigma = h_3(h_2(sk), M) \in Z_q$, where $h_2 : G \rightarrow Z_q$ and $h_3 : Z_q \rightarrow Z_q$ are hash functions.
end procedure $SignM$ **algorithm**

procedure $EncM$ **algorithm**
Input: The shared key (sk), the message to be transmitted (M), and the set of public parameters (pub).
Output: The ciphertext (τ) for M.
Step 1: V_A computes $\tau = h_2(sk) + M \bmod q$.
end procedure $EncM$ **algorithm**

$EncM$ algorithms for BMBP and ASBP are similar, except that the hash functions used in these two protocols are different (i.e., BMBP uses h_0, while ASBP uses h_2). This is because $sk \in G_T$ in BMBP, while $sk \in G$ in ASBP, as discussed in Algorithm 4 and Algorithm 5. However, this will not affect the performance of these two protocols, since the computation cost of hash function can be omitted as shown in Section V.

Step 3: Upon receiving (τ, σ, OTI_A, OTI_B, $para_A$, $para_B$) from V_A, V_B first extracts the temporary identities (TID_A and TID_B) in BMBP (or V_A and V_B in ASBP) from the one-time identities (OTI_A, OTI_B) and the corresponding parameters ($para_A, para_B$), using the following $Extr$ algorithm. The $Extr$ algorithms for BMBP and ASBP are defined in Algorithm 8 and Algorithm 9, respectively.

From Algorithm 8 and Algorithm 9, it can be seen that, the constructions of $Extr$ for BMBP and ASBP are different in two ways. First, when computing t, BMBP uses one pairing and one modular exponentiation, while ASBP uses only one modular exponentiation. Therefore, the total computation cost of ASBP in the $Extr$ algorithm is much lower than that of BMBP. Second, when computing r_A and r_B, the hash functions used in these two protocols are different (i.e., BMBP uses h_0, while ASBP uses h_2). This will not affect the performance of these two protocols, as discussed above in Algorithm 6 and Algorithm 7.

Step 4: After extracting temporary identities in BMBP (or real identities in ASBP), V_B generates the shared key sk using the following $GenkB$ algorithm, and then decrypts and verifies the message using sk and the following $VerM$ and $DecM$ algorithms, respectively. The message-receiving processes are defined in Algorithm 10 and Algorithm 11, respectively.

Algorithm 8: $Extr$ algorithm for BMBP.

procedure $Extr$ **algorithm**
Input: V_B's private key (sk_B), the nonce generated by V_B during the neighbor discovery phase (N_B), one-time identities (OTI_A and OTI_B) and their corresponding parameters ($para_A$ and $para_B$), and the set of public parameters (pub).
Output: The two temporary identities (TID_A and TID_B).
Step 1: V_B computes $t = e(sk_B, OTI_B^{N_B}) \in G_T$.
Step 2: V_B computes $r_A = para_A - h_0(t)$.
Step 3: V_B computes $r_B = para_B - h_0(t)$.
Step 4: V_B checks $OTI_B \stackrel{?}{=} g^{r_B}$. If this equation does not hold, V_B aborts because the message is not sent to it.
Step 5: V_B computes $TID_A = OTI_A^{r_A^{-1}}$.
end procedure $Extr$ **algorithm**

Algorithm 9: $Extr$ algorithm for ASBP.

procedure $Extr$ **algorithm**
Input: V_B's private key (sk_B), one-time identities (OTI_A and OTI_B) and their corresponding parameters ($para_A$ and $para_B$), and the set of public parameters (pub).
Output: The two real identities (V_A and V_B).
Step 1: V_B computes $t = g^{sk_B}$.
Step 2: V_B computes $r_A = para_A - h_2(t)$.
Step 3: V_B computes $r_B = para_B - h_2(t)$.
Step 4: V_B checks $OTI_B \stackrel{?}{=} g^{r_B}$. If this equation does not hold, V_B aborts because the message is not sent to it.
Step 5: V_B computes $V_A = OTI_A^{r_A^{-1}}$.
end procedure $Extr$ **algorithm**

Algorithm 10: Message-receiving processes of BMBP.

procedure $GenkB$ **algorithm**
Input: V_B's private key (sk_B), V_A's temporary identity (TID_A), V_B's temporary identity (TID_B), V_B's nonce generated in the neighbor discovery phase (N_B), and the set of public parameters (pub).
Output: A shared key (sk).
Step 1: V_B computes $sk = e(TID_A^{N_B}, sk_B) \in G_T$.
end procedure $GenkB$ **algorithm**

procedure $VerM$ **algorithm**
Input: The shared key (sk), the received message (M), the signature (σ) for M, and the set of public parameters (pub).
Output: T if M is not tampered. Or F otherwise.
Step 1: V_B computes $\sigma' = h_3(h_0(sk), M) \in Z_q$, where $h_0 : G_T \rightarrow Z_q$ and $h_3 : Z_q \rightarrow Z_q$ are hash functions.
Step 2: V_B checks $\sigma' \stackrel{?}{=} \sigma$. If this equation holds, it returns T. Otherwise, it returns F.
end procedure $VerM$ **algorithm**

procedure $DecM$ **algorithm**
Input: The shared key (sk), the ciphertext (τ), and the set of public parameters (pub).
Output: The plaintext (M).
Step 1: V_B computes $M = \tau - h_0(sk) \bmod q$.
end procedure $DecM$ **algorithm**

Algorithm 11: Message-receiving processes of ASBP.

procedure $GenkB$ **algorithm**
Input: V_B's private key (sk_B), V_A's real identity (V_A), V_B's real identity (V_B), and the set of public parameters (pub).
Output: A shared key (sk).
Step 1: V_B computes $h_1(V_A) = a_1 \cdots a_n$, where $h_1 : G \rightarrow \{0,1\}^n$ is a hash function, and $a_1, \cdots, a_n \in \{0,1\}$.
Step 2: V_B computes $sk = \left(\prod_{i=1}^{n} pk_{GW\,i}^{a_i}\right)^{sk_B} \in G$.
end procedure $GenkB$ **algorithm**

procedure $VerM$ **algorithm**
Input: The shared key (sk), the received message (M), the signature (σ) for M, and the set of public parameters (pub).
Output: T if M is not tampered. Or F otherwise.
Step 1: V_B computes $\sigma' = h_3(h_2(sk), M) \in Z_q$, where $h_3 : Z_q \rightarrow Z_q$ is a hash function.
Step 2: V_B checks $\sigma' \stackrel{?}{=} \sigma$. If this equation holds, it returns T. Otherwise, it returns F.
end procedure $VerM$ **algorithm**

procedure $DecM$ **algorithm**
Input: The shared key (sk), the ciphertext (τ), and the set of public parameters (pub).
Output: The plaintext (M).
Step 1: V_B computes $M = \tau - h_2(sk) \bmod q$. **end procedure** $DecM$ **algorithm**

Comparing $GenkB$ in Algorithm 10 and Algorithm 11, we can see that, the $GenkB$ in BMBP uses one pairing and one modular exponentiation, while that in ASBP uses only one modular exponentiation and several modular multiplications. Therefore, the total computation cost of ASBP in the $GenkB$ algorithm is much lower than that of BMBP. In addition, $sk \in G_T$ in BMBP, while $sk \in G$ in ASBP. This will result in different verification and decryption constructions of these two protocols as shown below.

Comparing $VerM$ and $DecM$ in Algorithm 10 and Algorithm 11, it can be seen that, the constructions of $VerM$ and $DecM$ algorithms for BMBP and ASBP are similar, except that the hash functions used in these two protocols are different (i.e., BMBP uses h_0, while ASBP uses h_2). This is because $sk \in G_T$ in BMBP, while $sk \in G$ in ASBP, as discussed in Algorithm 4 and Algorithm 5. However, this will not affect the performance of these two protocols, since the computation cost of hash function can be omitted as shown in Section V.

At the end of the data transmission phase, V_B accepts the message (M) sent from V_A.

Once V_A and V_B succeed in the above four steps, they can simplify the following data transmission processes as follows. First, V_A and V_B can generate subsequent one-time identities for anonymity in a simple way. For example, they can generate new one-time identity from the old one as $OTI_{new} = h(sk, OTI_{old})$, where $h : Z_q \rightarrow Z_q$ is a hash function. Second, V_A and V_B can generate sk as $sk_{new} = h(sk_{old})$. By doing so, the subsequent data transmission process can be quite light weight.

From the above initialization phase, it can be seen that, GW does not generate public keys for underwater vehicles. Therefore, the certificate management for public keys is avoided. Moreover, during the data transmission phase, V_A and V_B needs not consult GW for generating the shared key. Therefore, both BMBP and ASBP are suitable for underwater environment which lacks of security infrastructure.

From the above neighbor discovery phase, it can be seen that, V_A and V_B use temporary identities for establishing communications between them, and their real identities are not exposed. Therefore, BMBP has the zero-knowledge feature. In contrast, in ASBP, V_A and V_B must know real identities of each other before communication. Therefore, ASBP has no zero-knowledge feature.

From the above data transmission phase, it can be seen that, V_A and V_B use one-time identities for transmitting each message, and the adversary cannot extract real identities. Moreover, this adversary will be confused, and make the mistake that there are a lot of underwater vehicles. Therefore, both BMBP and ASBP have the anonymity feature.

From the above data transmission phase, it can be seen that, the message transmitted between V_A and V_B is signed and encrypted. And the adversary cannot decrypt and tamper the message (M). Therefore, both BMBP and ASBP can provide integrity and confidentiality protections for transmitted message.

From the above protocols, it can be seen that, V_A and V_B do not transmit additional messages for authentication and key establishment before data transmission. Therefore, both BMBP and ASBP can achieve the non-interactivity goal.

D. Further Discussions on the Two Protocols

The length of nonce: As illustrated in RFC2617 [35], a 64-bit nonce is likely more than sufficient for most practical purposes. Therefore, the length of nonces used in these two protocols can be further specified as 64-bit.

Reliability: In our protocols, vehicles use only one message for providing all the security functions, namely authentication, key establishment, data integrity protection and data confidentiality protection. However, this message may be lost. This is the reliability issue. To handle this issue, a typical method is to define an acknowledgement message. That is, when the receiver gets a message, it returns this message to acknowledge the reception of the transmitted message.

IV. Security Analysis

We analyze the security of our scheme with respect to the design goals given in Section III-B (i.e., anonymity, zero-knowledge, confidentiality and integrity).

Anonymity: As shown in Section III-C, V_A and V_B use one-time identities during data transmission. Therefore, the anonymity goal is to ensure that the adversary cannot extract real identities from transmitted data (τ, σ, OTI_A,

$OTI_B, para_A, para_B$). In Section IV-A, we'll prove that both BMBP and ASBP can achieve the anonymity goal.

Zero-knowledge: As shown in subsection III-C, V_A and V_B use temporary identities for discovering neighbors. Therefore, this goal is to ensure that V_A and V_B cannot extract real identities of each other from temporary identities. In Section IV-B, we'll prove that BMBP can achieve the zero-knowledge goal.

Confidentiality and integrity: At the same time, as shown in Section III-C, the transmitted message (M) is signed and encrypted using light-weight symmetric-key-based cryptographic primitives (hash function and modular addition), whose security is obvious. Therefore, the confidentiality and integrity goals are to ensure that the adversary cannot forge or extract the shared key (sk) from this protocol. In Sections IV-C and IV-D, we'll analyze the confidentiality and integrity of sk in both BMBP and ASBP, respectively.

Random oracle model [34] is a popular model for security analysis nowadays, which assumes the adversary can query oracles for getting some information of the cryptographic system before establishing an attack. In the underwater environment, it is possible for the adversary to get some information before establishing an attack, due to the openness of underwater wireless channel. Therefore, it is reasonable to analyze the security of our protocols in the random oracle model.

Our security analysis is comprised of three parts: security assumption, adversary model and security proof. Security assumption is a well-known mathematical problem which cannot be efficiently solved. Adversary model defines the adversary's activities. Security proof shows that the adversary cannot efficiently attack our scheme. Otherwise, we can use the adversary for solving the mathematical problem. In other words, the existence of this adversary contradicts the security assumption. So, if the security assumption holds, the adversary does not exist.

Security Assumption

Difficult mathematical problems on G:

1) The "Computational Diffie-Hellman (CDH)" problem [36]. For randomly distributed unknown $x, y \in Z_q$, one wants to compute g^{xy} from (g, g^x, g^y).

2) The "Bilinear Diffie-Hellman (BDH)" problem [37]. For randomly distributed unknown $x, y, z \in Z_q$, one wants to compute $e(g,g)^{xyz}$ from (g, g^x, g^y, g^z).

3) The "Discrete Logarithm (DL)" problem [38]. For randomly distributed unknown $x \in Z_q$, one wants to compute x from (g, g^x).

A. Anonymity

To achieve the anonymity goal, it must be ensured that the adversary between the two underwater vehicles cannot extract their real identities (i.e., V_A and V_B) from our protocols. This adversary ($\mathcal{A}$) holds pub, and can interact with a challenger ($\mathcal{C}$) as follows:

1) *Initialization:* $\mathcal{C}$ generates a set of public parameters (pub), and sends it to $\mathcal{A}$.

2) $GenTID$ *query:* $\mathcal{A}$ issues a series of $GenID$ queries to $\mathcal{C}$, and $\mathcal{C}$ returns TID_{ID} and $PK_{N_{ID}}$.

3) $Anony$ *query:* $\mathcal{A}$ issues a series of $Anony$ queries to $\mathcal{C}$, and $\mathcal{C}$ returns the one-time identities and the corresponding parameters (i.e., $OTI_A, OTI_B, para_A, para_B$).

4) *Extraction:* Given $(TID_A, TID_B, PK_{N_A}, PK_{N_B}, OTI_A, OTI_B, para_A, para_B, pub)$, the potential adversary $\mathcal{A}$ can extract V_A and V_B.

Then, for BMBP, we have

Theorem 1: If $\mathcal{A}$ can extract V_A and V_B from $(TID_A, TID_B, PK_{N_A}, PK_{N_B}, OTI_A, OTI_B, para_A, para_B, pub)$ with the probability ϵ in time t, $\mathcal{C}$ can solve the CDH problem with the probability $\epsilon' = \epsilon$ in time $t' = t + T_i$, where T_i is the time cost of inversion.

Proof: Refer to Appendix A.

Finally, for ASBP, we have

Theorem 2: If $\mathcal{A}$ can extract V_A and V_B from $(TID_A, TID_B, PK_{N_A}, PK_{N_B}, OTI_A, OTI_B, para_A, para_B, pub)$ with the probability ϵ in time t, $\mathcal{C}$ can solve the CDH problem with the probability $\epsilon' = \epsilon$ in time $t' = t + 2T_h + (\frac{n}{2} - 1)T_{prod} + T_{add} + T_m$, where T_h is the time cost of hash function, T_{prod} is the time cost of modular product, T_{add} is the time cost of modular addition, n is the number of public keys of the GW, and T_m is the time cost of modular exponentiation.

Proof: Refer to Appendix B.

B. Zero-Knowledge

As discussed in Section III, only BMBP has the zero-knowledge feature. To achieve the zero-knowledge goal, it must be ensured that one vehicle (e.g., V_A) cannot extract the real identity of the other (e.g., V_B). Therefore, the adversary is the vehicle ($e.g. V_A$) who wants to extract V_B. This adversary ($\mathcal{A}$) holds (sk_A, N_A, pub), and can interact with a challenger ($\mathcal{C}$) as follows:

1) *Initialization:* $\mathcal{C}$ generates a set of public parameters (pub), and sends it to $\mathcal{A}$.

2) $GenTID$ *query:* $\mathcal{A}$ issues a series of $GenID$ queries to $\mathcal{C}$, and $\mathcal{C}$ returns TID_{ID} and $PK_{N_{ID}}$.

3) *Extraction:* Given $(TID_A, TID_B, PK_{N_A}, PK_{N_B}, N_A, sk_A, pub)$, the potential adversary $\mathcal{A}$ can extract V_B.

For this adversary, we have

Theorem 3: If $\mathcal{A}$ can extract V_B from $(TID_A, TID_B, PK_{N_A}, PK_{N_B}, N_A, sk_A, pub)$ with the probability ϵ in time t, $\mathcal{C}$ can solve the CDH problem with the probability $\epsilon' = \epsilon$ in time $t' = t + T_i$, where T_i is the time cost of inversion.

Proof: Refer to Appendix C.

C. Confidentiality of sk

To achieve the confidentiality and integrity goals described in Section III, it must be ensured that the adversary between the two underwater vehicles cannot extract the shared key (sk) from this protocol. The adversary model is similar to that defined in Section IV.I, except that the potential adversary $\mathcal{A}$ outputs sk from $(TID_A, TID_B, PK_{N_A}, PK_{N_B}, OTI_A, OTI_B, para_A, para_B, pub)$ during the extraction process.

Moreover, for the bilinear-map-based protocol, we assume that the challenger knows $(V_A = g^{x_1}, V_B = g^{x_2})$. And prove

that, the adversary can establish an efficient attack on our protocol only with a negligible probability. Therefore, without knowing $(V_A = g^{x_1}, V_B = g^{x_2})$, it will be harder to attack our protocol. Then, we have

Theorem 4: If $\mathcal{A}$ can extract sk from $(TID_A, TID_B, PK_{N_A}, PK_{N_B}, OTI_A, OTI_B, para_A, para_B, pub)$ with the probability ϵ in time t, $\mathcal{C}$ can solve the BDH problem with the probability $\epsilon' = \epsilon$ in time $t' = t + T_i + T_m + T_{prod}$, where T_i is the time cost of inversion, T_m is the time cost of modular exponentiation, and T_{prod} is the time cost of modular product.

Proof: Refer to Appendix D.

Finally, for the algebraic-signature-based protocol, we have

Theorem 5: If $\mathcal{A}$ can extract sk from $(TID_A, TID_B, PK_{N_A}, PK_{N_B}, OTI_A, OTI_B, para_A, para_B, pub)$ with the probability ϵ in time t, $\mathcal{C}$ can solve the CDH problem with the probability $\epsilon' = \epsilon$ in time $t' = t$.

Proof: Refer to Appendix E.

D. Integrity of sk

To achieve the integrity and confidentiality goals described in Section III, it must be ensured that the adversary between the two underwater vehicles cannot tamper the shared key (sk) from this protocol. The adversary model is similar to that defined in Section IV-A, except that the potential adversary $\mathcal{A}$ tampers sk to $sk' \neq sk$ from $(TID_A, TID_B, PK_{N_A}, PK_{N_B}, OTI_A, OTI_B, para_A, para_B, pub)$.

For the bilinear-map-based protocol, the proof is similar to Theorem 4 as summarized as follows. If $\mathcal{A}$ can tamper sk to $sk' \neq sk$, $\mathcal{C}$ can runs this adversary with the same parameters in Theorem 4 to get $sk' \neq sk$. Then $\mathcal{C}$ can solve the BDH as $e(g,g)^{xyz} = sk'^{(x_1 x_2)^{-1}}$.

For the algebraic-signature-based protocol, the proof is similar to Theorem 5 too. Thus, omitted.

E. Further Discussions on the Security of These Two Protocols

Replay attacks: In this paper, we mainly focus on the anonymity and zero-knowledge features in underwater environment. However, there may be some traditional security issues that have not been discussed such as replay attacks. Traditional method for preventing replay attacks is to use time stamps, where the receiver treats an expired message as a replayed message and discards it. In contrast, in our two newly designed protocols, replay attacks can be avoided using one-time identities. This is illustrated by the following example. When getting a message, the receiver just checks identities in the message. If the identities have been used previously, the receiver can simply treat it as a replayed message and discard it. This is another reason why the one-time identities cannot be used twice.

Security of the algebraic-signature-based protocol: The algebraic signature is vulnerable to the following attack. For example, in the algebraic signature technique described in Section III-A, if $n = 2$ and two underwater vehicles (e.g., V_A and V_B) are broken, the adversary will get two equations $sk_A = x_{1A} sk_{GW1} + x_{2A} sk_{GW2}$ and $sk_B = x_{1B} sk_{GW1} + x_{2B} sk_{GW2}$, where $x_{1A}, x_{2A}, x_{1B}, x_{2B}$ are hashed bits described in the $Gensk$ algorithm of Section III-D. In these two equations, since the adversary knows $sk_A, sk_B, x_{1A}, x_{2A}, x_{1B}, x_{2B}$, it can easily compute sk_{GW1} and sk_{GW2}. That is, the adversary can compute private keys of the GW (sk_{GW}). In this case, the adversary can do anything it wants to do and the cryptographic system is defeated. However, in the underwater environment, this sort of attacks does not work as illustrated below. First, for the security level $n = 112$, the adversary has to break more than 112 underwater vehicles to defeat our protocol, which is very hard. Second, in the underwater environment, it is difficult to break vehicles internally. Third, to avoid such attacks, the GW can increase n, or split vehicles into groups where the number of vehicles in each group is less than n, resulting in additional storage costs.

V. Performance Evaluation

Among all data transmission protocols for underwater wireless communications [22-27,31], SET-IBS and SET-IBOOS are similar to our newly designed protocols (BMBP and ASBP), because they are non-interactive identity-based data transmission protocols, while other data transmission protocols [22–27] lack the non-interactivity feature and are based on security infrastructure. Therefore, in this section, we mainly compare BMBP and ASBP with SET-IBS and SET-IBOOS [31].

As described in Section III-B, to achieve high performance, it must be ensured that the computation, communication and storage costs are low. In Section V-A, we first compare the computation costs of these four data transmission protocols, namely BMBP, ASBP, SET-IBS and SET-IBOOS. Then, in Section V-B and Section V-C, we compare communication and storage costs of these four protocols, respectively. Finally, we implement these two newly designed protocols to check their validity in Section V-D, and present an overall comparison for these four protocols (BMBP and ASBP with SET-IBS and SET-IBOOS) in Section V-E.

A. Comparison of Computation Costs

Computation cost is defined as the total time cost of cryptographic algorithms. In this subsection, we first investigated the time costs of basic cryptographic algorithms (i.e., hash function, inversion, modular multiplication, modular exponentiation and bilinear pairing), and then compared the computation costs of BMBP, ASBP, SET-IBS and SET-IBOOS.

To investigate the time costs of basic cryptographic algorithms, we conducted our experiments on a CENTOS operating system installed on an Intel i7 processor. Cryptographic libraries used in our experiments are PBC [39] and OPENSSL [40]. To achieve the 112-bit security level, we chose the 224-bit elliptic curve groups [40]. In this case, the modular exponentiation and modular multiplication algorithms used in our experiments are the same as elliptic curve scalar multiplication and point addition algorithms, respectively. Finally, bilinear pairing parameter used is the type d224 parameter [39], while hash function is SHA256 [40].

Table III shows the time costs of basic cryptographic algorithms, which are the average results of running basic cryptographic algorithms for 10,000 times. In Table III, the time cost

TABLE III
BASIC CRYPTOGRAPHIC ALGORITHMS (UNIT: μS)

T_{bp}	T_m	T_i	T_{SHA256}	T_{prod}
10631.0	829.0	0.5	0.6	0.1

of bilinear pairing is denoted by T_{bp}, the time cost of modular exponentiation is denoted by T_m, the time cost of inversion is denoted by T_i, the time cost of SHA1 is denoted by T_{SHA256}, and the time cost of modular multiplication is denoted by T_{prod}.

From Table III, we can draw the following conclusions:

1) Compared with the bilinear pairing and modular exponentiation algorithms, the time costs of inversion, SHA1 and modular multiplication can be omitted. This is because $T_m/T_{SHA256} \approx 1.4 \times 10^3$, $T_{bp}/T_{SHA256} \approx 1.8 \times 10^4$, $T_{SHA256}/T_i = 1.2$, and $T_{SHA256}/T_{prod} = 6$.

2) The time cost of modular exponentiation is much lower than that of bilinear pairing, because $T_{bp}/T_m \approx 1.3 \times 10^1$.

The above two conclusions show that the bilinear pairing algorithm is much more time-consuming than other algorithms. Therefore, to avoid using bilinear map, our algebraic-signature-based protocol can reduce the computation cost significantly.

Then, we computed the computation costs of BMBP, ASBP, SET-IBS and SET-IBOOS from Table III. The results are shown in Table IV, where the computation cost of V_A is denoted by T_{VA}, the computation cost of V_B is denoted by T_{VB}, and the total computation cost is denoted by T_t. Note that we only take the bilinear pairing and modular exponentiation algorithms into account in Table IV, because the modular multiplication and hash function can be omitted as discussed in Table III.

From Table IV, we can draw the following conclusions:

1) On V_A's side, the computation costs of ASBP and SET-IBOOS are around 10^{-1} to those of BMBP and SET-IBS. This is because $2.49/24.58 \approx 1.0 \times 10^{-1}$, $3.32/24.58 \approx 1.4 \times 10^{-1}$, $2.49/13.95 \approx 1.8 \times 10^{-1}$, $3.32/13.95 \approx 2.4 \times 10^{-1}$.

2) On V_B's side, the computation costs of ASBP and SET-IBOOS are around 10^{-1} to those of BMBP and SET-IBS too. This is because $3.32/24.58 \approx 1.4 \times 10^{-1}$, $2.49/24.58 \approx 1.0 \times 10^{-1}$, $3.32/22.09 \approx 1.5 \times 10^{-1}$ and $2.49/22.09 \approx 1.1 \times 10^{-1}$.

3) The total computation costs of ASBP and SET-IBOOS are around 10^{-1} to those of BMBP and SET-IBS. This is because $5.80/49.16 \approx 1.2 \times 10^{-1}$ and $5.80/36.04 \approx 1.6 \times 10^{-1}$.

4) The computation costs of BMBP and SET-IBS are at the same level. This is because $24.58/13.95 \approx 1.8 \times 10^0$, $24.58/22.09 \approx 1.1 \times 10^0$ and $49.16/36.04 \approx 1.4 \times 10^0$.

5) The computation costs of ASBP and SET-IBOOS are at the same level. This is because $3.32/2.49 = 1.3 \times 10^0$.

The above five conclusions show that, the computation costs of ASBP and SET-IBOOS can be much lower than those of BMBP and SET-IBS. Moreover, from Table IV, it can be seen that the computation costs of all four protocols are at the millisecond level. Therefore, all four protocols are feasible to be deployed in underwater environment.

B. Comparison of Communication Costs

In underwater wireless environment, the main concern for communication cost is number of messages. As discussed in Sections I and II, if additional messages are transmitted for authentication and key establishment and they are lost, subsequent messages may be delayed. Therefore, it is desired to use one message for providing all these functions. In both BMBP and ASBP, vehicles do not need transmit additional messages. Therefore, our protocols are highly efficient. Similarly, in SET-IBS and SET-IBOOS, vehicles do not need transmit additional messages either. The above discussion shows that all the four protocols can achieve the non-interactivity goal.

On the other hand, in other data transmission protocols for underwater wireless communications [22-27], vehicles have to transmit additional messages for authentication and key establishment. Therefore, these protocols [22-27] cannot achieve the non-interactivity goal.

Message length is another concern for communication cost. In both protocols, V_A transports $(\tau, \sigma, OTI_A, OTI_B, para_A, para_B)$ to V_B. In all these parameters, τ, σ, $para_A$, and $para_B$ are integers in the finite field Z_q whose length is $224-$bit, while OTI_A and OTI_B are elliptic curve points whose length is $224 \times 2 = 448-$bit. Therefore, the communication costs of BMBP and ASBP are both $224 \times 4 + 448 \times 2 = 1792-$bit $= 224$ bytes. The communication costs of SET-IBS and SET-IBOOS are around 200 bytes too. Therefore, the communication costs of all four protocols are at the same level (i.e., around 200 bytes), and can satisfy the underwater environment.

C. Comparison of Storage Costs

The storage costs on V_A and V_B refer the cryptographic parameters received from GW in the initialization phase. In both BMBP and ASBP, V_A (or V_B) stores (V_A, sk_A, pub) (or (V_B, sk_B, pub)). The difference is that, in BMBP, V_A (or V_B) stores one public key of the GW whose length is $448 - bit = 56$ bytes, while in ASBP, V_A (or V_B) stores n public key of the GW whose length is $448n - bit$. For $n = 112$, the storage cost of ASBP is around $50176 - bit = 6.3$ kilobytes. Therefore, the storage cost of ASBP is much higher than BMBP. For SET-IBS and SET-IBOOS, their storage costs are similar to that of BMBP.

In addition, in all four protocols, the storage costs on V_A and V_B are fixed, and independent of the number of vehicles and messages. Therefore, all four protocols can still work well when transmitting a lot of messages.

D. Implementation of the Newly Designed Protocols

To check the validity of the newly designed two protocols, we implemented them. Our experimental environment is similar to that used in Section V-A, except that three computers are employed, acting as the GW, V_A and V_B, respectively. These three computers are connected over the 1 Gbps Ethernet. As an instance, we set the length of transmitted message to be $100-$bit. Finally, we get the data transmitting time of BMBP $T_{BMBP} \approx 50$ ms and that of ASBP $T_{ASBP} \approx 6$ ms. These experimental results are similar to the results theoretically computed in Table IV. Therefore, both protocols are feasible for real applications.

TABLE IV
COMPUTATION COSTS (UNIT: MS)

	BMBP	ASBP	SET-IBS	SET-IBOOS
T_{VA}	$2T_{bp}+4T_m=24.58$	$3T_m=2.49$	$T_{bp}+4T_m=13.95$	$4T_m=3.32$
T_{VB}	$2T_{bp}+4T_m=24.58$	$4T_m=3.32$	$2T_{bp}+T_m=22.09$	$3T_m=2.49$
T_t	$4T_{bp}+8T_m=49.16$	$7T_m=5.80$	$3T_{bp}+5T_m=36.04$	$7T_m=5.80$

TABLE V
OVERALL COMPARISON

	BMBP	ASBP	SET-IBS	SET-IBOOS
Non-interactivity	√	√	√	√
Identity-based	√	√	√	√
Integrity	√	√	√	√
Anonymity	√	√	×	×
Zero-knowledge	√	×	×	×
Confidentiality	√	√	×	×
Computation Cost(ms)	10^1	10^0	10^1	10^0
number of messages	1	1	1	1
Storage cost($byte$)	10^2	10^3	10^2	10^2

E. Overall Comparison

To further evaluate the performance of our newly designed protocols, we present an overall comparison of the four protocols (BMBP, ASBP, SET-IBS and SET-IBOOS), as shown in Table V.

From Table V, it can be seen that, all four protocols are non-interactive identity-based data transmission protocols which can provide integrity protection for transmitted messages. And the performance of the four protocols are similar. However, our protocols can achieve more security goals, namely anonymity, zero-knowledge and confidentiality, which are very important for underwater wireless communications.

VI. CONCLUSION

The underwater environment is quite different from terrestrial environment. First, the underwater environment lacks traditional security infrastructures such as public key infrastructure. So the authentication and key management processes are hard. Second, the underwater wireless channel is poor. So the transmitted messages will be frequently lost. Third, in the underwater environment, vehicles may know nothing about each other and do not want to expose their real identities. So it is quite hard to design security protocols. All these problems indicate that designing a security protocol for underwater wireless communication systems is a challenging work.

In this paper, we have analyzed the security problems in underwater wireless communication systems, and classified a set of six design goals for secure data transmission protocols in underwater environment. Moreover, we have proposed two data transmission protocols. One is based on bilinear map which is suitable for underwater vehicles with high computation cost and low storage cost, and the other is based on algebraic signature which is suitable for underwater vehicles with low computation cost and high storage cost. The security analysis and experimental results show that the proposed approaches are secure, and feasible for real-world applications.

However, there are several more open issues remaining to be solved. First, when the keying materials expire, how to update secret keys and public parameters for underwater vehicles? Second, when a vehicle leaves the underwater wireless communication system, how to revoke its keying materials? Third, when there are many vehicles, these vehicles may be organized into several clusters, and some vehicles may act as cluster headers. In this case, how to design a data transmission protocol with the existence of cluster headers? Therefore, in the future, we'll further address these open issues.

APPENDIX

A. Proof of Theorem 1

Proof: Let g be the generator of G. Given (g, g^x, g^y) for randomly distributed unknown $x, y \in Z_q$, $\mathcal{C}$ can compute g^{xy} as follows.

1) $\mathcal{C}$ runs $\mathcal{A}$ with $(TID_A = g^x, TID_B, PK_{N_A} = g^y, PK_{N_B}, OTI_A, OTI_B, para_A, para_B, pub)$ to get (V_A, V_B). The probability of success is ϵ and the time is t.

2) From the $GenTID$ algorithm, we get $TID_A = V_A^{N_A} \Rightarrow g^x = V_A^{N_A} \Rightarrow g^{N_A^{-1}x} = V_A \Rightarrow PK_{N_A}^{-x} = V_A \Rightarrow g^{-xy} = V_A \Rightarrow g^{xy} = V_A^{-1}$. The probability of success is 1 and the time cost is T_i.

3) The probability is the product of the probabilities in the above two steps $\epsilon' = \epsilon \times 1 = \epsilon$, and the time cost is the addition of the time costs in the above two steps $t' = t + T_i$.

B. Proof of Theorem 2

Proof: Let g be the generator of G. Given (g, g^x, g^y) for randomly distributed unknown $x, y \in Z_q$, $\mathcal{C}$ can compute g^{xy} as follows.

1) $\mathcal{C}$ runs $\mathcal{A}$ with $(TID_A, TID_B, PK_{N_A}, PK_{N_B}, OTI_A = g^x, OTI_B = g^y, para_A, para_B, pub)$ to get (V_A, V_B). The probability of success is ϵ and the time is t.

2) $\mathcal{C}$ computes $h_1(V_B) = b_1 \cdots b_n$. The probability of success is 1 and the time cost is T_h.

3) $\mathcal{C}$ computes $t = \prod_{i=1}^{n} pk_{GWi}^{b_i} \in G$. The probability of success is 1 and the time cost is $(\frac{n}{2} - 1)T_{prod}$ (Note that $b_i \in \{0, 1\}$ and the probability that $b_i = 1$ is $\frac{1}{2}$. Therefore, in the above equation $\prod_{i=1}^{n} pk_{GWi}^{b_i}$, there are only $\frac{n}{2} - 1$ modular products, because $pk_{GWi}^{b_i} = 1$ (for $b_i = 0$) and $pk_{GWi}^{b_i} = pk_{GWi}$ (for $b_i = 1$).).

4) $\mathcal{C}$ computes $r_B = para_B - h_2(t)$. The probability of success is 1 and the time cost is $T_{add} + T_h$.

5) $\mathcal{C}$ computes $g^{xy} = (OTI_A)^{r_B}$. The probability of success is 1 and the time cost is T_m.

6) The probability is the product of the probabilities in the above two steps $\epsilon' = \epsilon \times 1 \times 1 \times 1 \times 1 = \epsilon$, and the time cost is the addition of the time costs in the above two steps $t' = t + T_h + (\frac{n}{2} - 1)T_{prod} + T_{add} + T_h + T_m = t + 2T_h + (\frac{n}{2} - 1)T_{prod} + T_{add} + T_m$.

C. Proof of Theorem 3

Proof: Let g be the generator of G. Given (g, g^x, g^y) for randomly distributed unknown $x, y \in Z_q$, $\mathcal{C}$ can compute g^{xy} as follows.

1) $\mathcal{C}$ runs $\mathcal{A}$ with $(TID_A, TID_B = g^x, PK_{N_A}, PK_{N_B} = g^y, N_A, sk_A, pub)$ to get V_B. The probability of success is ϵ and the time is t.

2) From the $GenTID$ algorithm, we get $TID_B = V_B^{N_B} \Rightarrow g^x = V_B^{N_B} \Rightarrow g^{N_B^{-1}x} = V_B \Rightarrow PK_{N_B}^{-x} = V_B \Rightarrow g^{-xy} = V_B \Rightarrow g^{xy} = V_B^{-1}$. The probability of success is 1 and the time cost is T_i.

3) The probability is the product of the probabilities in the above two steps $\epsilon' = \epsilon \times 1 = \epsilon$, and the time cost is the addition of the time costs in the above two steps $t' = t + T_i$.

D. Proof of Theorem 4

Proof: Let g be the generator of G. Given (g, g^x, g^y, g^z) for randomly distributed unknown $x, y, z \in Z_q$, $\mathcal{C}$ can compute $e(g, g)^{xyz}$ as follows.

1) $\mathcal{C}$ runs $\mathcal{A}$ with $(TID_A, V_A = g^{x_1}, V_B = g^{x_2}, TID_B, PK_{N_A} = g^y, PK_{N_B} = g^z, OTI_A, OTI_B, para_A, para_B, pub, pk_{GW} = g^x)$ to get sk. The probability of success is ϵ and the time is t.

2) From the $GenKA$ algorithm, we get $sk = e(sk_A, TID_B^{N_A}) = e((V_A)^{sk_{GW}}, V_B^{N_A N_B}) = e(V_A, V_B)^{sk_{GW} N_A N_B} = e(g^{x_1}, g^{x_2})^{sk_{GW} N_A N_B} \Rightarrow e(g, g)^{xyz} = e(g, g)^{sk_{GW} N_A N_B} = sk^{(x_1 x_2)^{-1}}$. the probability of success is 1 and the time cost is T_i.

3) The probability is the product of the probabilities in the above two steps $\epsilon' = \epsilon \times 1 = \epsilon$, and the time cost is the addition of the time costs in the above two steps $t' = t + T_i + T_m + T_{prod}$.

E. Proof of Theorem 5

Proof: Let g be the generator of G. Given (g, g^x, g^y) for randomly distributed unknown $x, y \in Z_q$, $\mathcal{C}$ can compute g^{xy} as follows.

1) $\mathcal{C}$ runs $\mathcal{A}$ with $(TID_A, V_A = g^x, V_B = g^x, TID_B, PK_{N_A}, PK_{N_B}, OTI_A, OTI_B, para_A, para_B, pub)$ to get sk. The probability of success is ϵ and the time is t.

2) From the $GenKA$ algorithm, we get $sk = g^{sk_A sk_B} \Rightarrow g^{xy} = g^{sk_A sk_B} = sk$. the probability of success is 1 and the time cost is 0 (This is straightforward.).

3) The probability is the product of the probabilities in the above two steps $\epsilon' = \epsilon \times 1 = \epsilon$, and the time cost is the addition of the time costs in the above two steps $t' = t + 0 = t$.

REFERENCES

[1] X. Zhang *et al.*, "Underwater wireless communications and networks: Theory and application: Part 1," *IEEE Commun. Mag.*, vol. 53, no. 11, pp. 40–41, Nov. 2015.

[2] X. Zhang *et al.*, "Underwater wireless communications and networks: Theory and application: Part 2," *IEEE Commun. Mag.*, vol. 54, no. 2, pp. 30–31, Feb. 2016.

[3] R. W. L. Coutinho *et al.*, "On the design of green protocols for underwater sensor networks," *IEEE Commun. Mag.*, vol. 54, no. 10, pp. 67–73, Oct. 2016.

[4] A. Caiti *et al.*, "Secure cooperation of autonomous mobile sensors using an underwater acoustic network," *Sensors*, vol. 12, no. 2, pp. 1967–1989, Feb. 2012.

[5] D. Cooper *et al.*, "Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile," IETF, RFC5280, May 2008.

[6] P. Yee, "Updates to the internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile," IETF, RFC6818, Jan. 2013.

[7] M. Burmester *et al.*, "Secure and privacy-preserving, timed vehicular communication," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 10, no. 4, pp. 219–229, Sep. 2012.

[8] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. Adv. Cryptol.*, 1984, pp. 47–53.

[9] S. Misra *et al.*, "Jamming in underwater sensor networks: Detection and mitigation," *IET Commun.*, vol. 6, no. 12, pp. 2178–2188, Dec. 2012.

[10] M. Zuba *et al.*, "Vulnerabilities of underwater acoustic networks to denial-of-service jamming attacks," *Security Commun. Netw.*, vol. 8, no. 16, pp. 2635–2645, Nov. 2015.

[11] H. Kulhandjian *et al.*, "Securing underwater acoustic communications through analog network coding," in *Proc. 2014 11th Annu. IEEE Int. Conf. Sens., Commun., Netw.*, Jun. 2014, pp. 266–274.

[12] H. Zhang and Y. Dong, "Impulse response modeling for general underwater wireless optical MIMO links," *IEEE Commun. Mag.*, vol. 54, no. 2, pp. 56–61, Feb. 2016.

[13] Y. Luo *et al.*, "RSS-based secret key generation in underwater acoustic networks: Advantages, challenges, and performance improvements," *IEEE Commun. Mag.*, vol. 54, no. 2, pp. 32–38, Feb. 2016.

[14] Y. Liu, J. Jing, and J. Yang, "Secure underwater acoustic communication based on a robust key generation scheme," in *Proc. 9th IEEE Int. Conf. Signal Process.*, Nov. 2008, pp. 1838–1841.

[15] F. Hu, S. Wilson, and Y. Xiao, "Correlation-based security in time synchronization of sensor networks," in *Proc. 2008 Wireless Commun. Netw. Conf.*, Mar. 2008, pp. 2525–2530.

[16] F. Hu *et al.*, "Vertical and horizontal synchronization services with outlier detection in underwater acoustic networks," *Wireless Commun. Mobile Comput.*, vol. 8, no. 9, pp. 1165–1181, Nov. 2008.

[17] M. Xu *et al.*, "Cluster-based secure synchronization protocol for underwater wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 2014, no. 1, pp. 1–13, Apr. 2014.

[18] W. Wang *et al.*, "Visualisation of wormholes in underwater sensor networks: A distributed approach," *Int. J. Security Netw.*, vol. 3, no. 1, pp. 10–23, Jan. 2008.

[19] R. Zhang, and Y. Zhang, "Wormhole-resilient secure neighbor discovery in underwater acoustic networks," in *Proc. 29th IEEE INFOCOM*, Mar. 2010, pp. 2633–2641.

[20] G. Dini and A. L. Duca, "A secure communication suite for underwater acoustic sensor networks," *Sensors*, vol. 12, no. 11, pp. 15133–15158, Nov. 2012.

[21] R. W. L. Coutinho *et al.*, "Design guidelines for opportunistic routing in underwater networks," *IEEE Commun. Mag.*, vol. 54, no. 2, pp. 40–48, Feb. 2016.

[22] E. Souza *et al.*, "End-to-end authentication in under-water sensor networks," in *Proc. 2013 IEEE Symp. Comput. Commun.*, Jul. 2013, pp. 299–304.

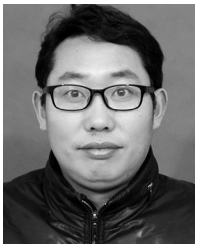
[23] S. Verma and Prachi, "A cluster based key management scheme for underwater wireless sensor networks," *Int. J. Comput. Network Inf. Security*, vol. 2015, no. 9, pp. 54–63, Aug. 2015.

[24] M. Ibragimov *et al.*, "CCM-UW security modes for low-band underwater acoustic sensor networks," *Wireless Pers. Commun.*, vol. 89, no. 2, pp. 479–499, Jul. 2016.

[25] Y. P. Kim *et al.*, "Data encryption and authentication mechanism based on block cipher mode for underwater acoustic sensor networks," in *Proc. 2012 Int. Conf. Inf. Sci. Technol.*, Jun. 2012, pp. 28–30.

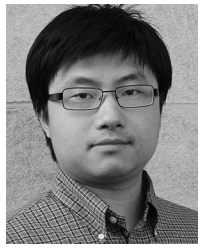
[26] M. Park, Y. Kim, and O. Yi, "Light weight authentication and key establishment protocol for underwater acoustic sensor networks," *The J. Korean Inst. Commun. Inf. Sci.*, vol. 39, no. 6, pp. 360–369, Jun. 2014.

[27] G. Ateniese G *et al.*, "SecFUN: Security framework for underwater acoustic sensor networks," in *Proc. MTS/IEEE Oceans*, May 2015, pp. 1–9.
[28] J. E. Kim *et al.*, "Security in underwater acoustic sensor network: Focus on suitable encryption mechanisms," in *Proc. Asia Simul. Conf.*, Oct. 2012, pp. 160–168.
[29] Y. Zhang *et al.*, "Node secure localization algorithm in underwater sensor network based on trust mechanism," *J. Comput. Appl.*, vol. 33, no. 5, pp. 1208–1211, Oct. 2013.
[30] G. Han *et al.*, "Management and applications of trust in wireless sensor networks: A survey," *J. Comput. Syst. Sci.*, vol. 80, no. 3, pp. 602–617, May 2014.
[31] H. Lu, J. Li, and M. Guizani, "Secure and efficient data transmission for cluster-based wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 3, pp. 750–761, Mar. 2014.
[32] D. Bonehand and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. Adv. Cryptol.*, Jan. 2001, pp. 213–229.
[33] T. J. E. Schwarz and E. L. Miller, "Store, forget, and check: Using algebraic signatures to check remotely administered storage," in *Proc. 26th IEEE Int. Conf. Distrib. Comput. Syst.*, Feb. 2006, p. 12.
[34] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proc. 1st ACM Conf. Comput. Commun. Security*, Jan. 1993, pp. 62–73.
[35] J. Franks *et al.*, "HTTP authentication: Basic and digest access authentication," IETF, RFC2617, Jun. 1999.
[36] F. Bao, R. H. Deng, and H. Zhu, "Variations of Diffie–Hellman problem," in *Proc. Int. Conf. Inf. Commun. Security*, Oct. 2003, pp. 301–312.
[37] R. Dutta, R. Barua, and P. Sarkar, "Pairing-based cryptographic protocols: A survey," *IACR Cryptol. ePrint archive*, Jul. 2004, pp. 1–45.
[38] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Rev.*, vol. 41, no. 2, pp. 303–332, Jun. 1999.
[39] B. Lynn, "PBC library manual 0.5.11," 2006. [Online]. Available: http://crypto.stanford.edu/pbc/manual/
[40] Openssl.org, "Openssl-1.0.1e.tar.gz," Feb. 2013. [Online]. Available: http://www.openssl.org/source/



Changsheng Wan received the B.S. degree in applied physics from University of Science and Technology of China, Hefei, China, in 1999, and the Ph.D. degree in physical electronics from University of Science and Technology of China, Hefei, in 2004. From July 2004 to March 2007, he was with Huawei Technology. Since April 2007, he has been with Southeast University as an Associate Professor. His research interests include the areas of network security, wireless communication, and data mining.

Vir Virander Phoha is a Professor in the College of Engineering and Computer Science, Syracuse University, Syracuse, New York, USA. His research interests include the areas of network security and biometrics.



Yuzhe Tang is an Assistant Professor in the College of Engineering and Computer Science, Syracuse University, Syracuse, New York, USA. His research interests include the areas of network security and multiparty computation.

Aiqun Hu is a Professor in the School of Information Science and Engineering, Southeast University, Nanjing, Jiangsu, China. His research interests include the areas of network security and wireless communication.