Securing Underwater Communications: Key Agreement based on Fully Hashed MQV

Angelo Capossele
Sapienza University of Rome
Dept. of Computer Science
capossele@di.uniroma1.it

Chiara Petrioli
Sapienza University of Rome
Dept. of Computer Science
petrioli@di.uniroma1.it
WSENSE s.r.l., Rome, Italy
chiara.petrioli@wsense.it

Gabriele Saturni
Sapienza University of Rome
Dept. of Computer Science
saturni@di.uniroma1.it

Daniele Spaccini
Sapienza University of Rome
Dept. of Computer Science
spaccini@di.uniroma1.it
WSENSE s.r.l., Rome, Italy
daniele.spaccini@wsense.it

Daniele Venturi
Sapienza University of Rome
Dept. of Computer Science
venturi@di.uniroma1.it

ABSTRACT

This paper concerns the implementation and testing of a protocol that two honest parties can efficiently use to share a common secret session key. The protocol, based on the Fully Hashed Menezes-Qu-Vanstone (FHMQV) key agreement, is optimized to be used in underwater acoustic communications, thus enabling secure underwater acoustic networking. Our optimization is geared towards obtaining secure communications without affecting network performance by jointly keeping security-related overhead and energy consumption at bay. Implementation and testing experiments have been performed with the SUNSET SDCS framework and its SecFUN extension using as hardware two submerged acoustic modems. Results show that our approach imposes a low computational burden to the underwater node, which implies low local energy consumption. This is due to the fact the FHMQV protocol is highly efficient resulting in a small number of operations with a low computation cost. In addition the use of elliptic curves allows to further reduce the computational overhead.

KEYWORDS

SecFUN, SUNSET, underwater acoustic sensor networks, underwater security, key agreement protocol

WUWNET'17, November 6–8, 2017, Halifax, NS, Canada

ACM ISBN 978-1-4503-5561-2/17/11…$15.00
https://doi.org/10.1145/3148675.3152760

1 INTRODUCTION

Interest and research activity around underwater acoustic sensor networks (UASNs) has constantly increased during recent years, driving a large number of applications, including scientific and commercial exploration, coastal protection and prediction of underwater natural disasters. In this evolving context, security risks and threats are ever more critical. The broadcasting nature of the UASN channel makes the data vulnerable to being modified, injected and eavesdropped. As a consequence, attacks could disrupt the regular network flow by creating loops, attracting or rejecting traffic in specific areas, partitioning the network or creating bad routes. Developing solutions for secure underwater communication and networking is therefore of paramount importance. However, the design cannot neglect the constraints imposed by the underwater channel. The use of acoustic transmissions introduces several challenges such as low data rate, variable and long propagation delays, significant variations in terms of link reliability over time and long interference range. Furthermore, sensor nodes are usually powered by batteries whose replacement can increase costs and complexity. Only recently, solutions have been proposed in the literature to enhance the security of UASNs as well as to cope with their challenges [2, 4], delivering security properties ranging from message confidentiality, to message authentication and integrity via digital signatures.

Secure communication mainly grounds its roots in the implementation of robust Key Management Protocols (KMPs). In fact, the security of secret keys is of paramount importance not only to guarantee data confidentiality but also as fundamental requirement to provide data authentication and integrity. Key agreement protocols based on symmetric cryptography are appealing for their efficiency in terms of both computation and communication. However, they lack flexibility and typically assume a pre-shared key (or a set of predefined keys in the case of random key pre-distribution) among network nodes. If an attacker compromises a node,

and therefore the pre-shared key, the security of the whole network would be compromised. On the other hand, the use of Public Key Cryptography (PKC) provides more flexible and secure authentication mechanisms. PKC schemes allow two or more peers to agree on a common secret key by establishing a secure and authenticated channel such that, if a node is compromised, only the secure channels of that node would be compromised. Feasibility of PKC techniques over UASN devices has been proven by recent implementation based on Elliptic Curve Cryptography (ECC), even over severely-constrained devices [2, 4]. Unfortunately, key agreement protocols based on PKC typically require the exchange of explicit certificates (e.g., Elliptic Curve Diffie Hellman uses X.509 certificates) having unfeasible size for UASNs (hundred of bytes). One of the most promising directions to overcome this drawback is the adoption of implicit certificates, which binds the identity of a node and its public key within a single data structure, and which can certify the authenticity of such a relation without an explicit signature [17]. Thanks to their small size, the implicit certificate are one of the best choice for key agreement in the underwater scenarios.

In this paper, we propose an implementation of a key agreement protocol based on the Fully Hashed Menezes-Qu-Vanstone (FHQMV) [16] and the Elliptic Curve based Qu-Vanstone (ECQV) implicit certificate scheme. To the best of our knowledge, this is the first work evaluating the suitability of key agreement protocols based on implicit certificates for UASNs.

The protocol has been implemented in SUNSET Software Defined Communication Stack (SDCS), a framework to simulate, emulate and test in real-life communication protocols at all level of the protocol stack [14]. To demonstrate the feasibility of our solution, we tested it with real acoustic modems. We experimentally assess the performance of our implementation, showing that the computational and networking overhead, with related energy consumption, are minimal.

The rest of the paper is organized as follows. Previous works on underwater security are summarized in Section 2. In Sections 3 and 4, we present the key agreement protocol and its optimizations for UASNs. The system architecture used for the protocol implementation is described in Section 5, while the performance results are shown in Section 6. Finally, Section 7 concludes the paper.

2 RELATED WORK

Challenges related to UASNs security have been increasingly explored during recent years. Several works, including [6, 8, 12] explore security issues in UASNs by analyzing attacks, threats and possible countermeasures and highlighting the importance of developing security solutions tailored to UASNs characteristics. In the attempt of delivering energy-friendly security solutions, the work proposed in [19] explicits the native support of Elliptic Curve Cryptography (ECC) by exploiting the Digital Signal Processors of acoustic modems to speed-up the computation of cryptographic operations. In [18], authors present a comparison among different digital signature schemes in terms of energy efficiency and suitability for both underwater and terrestrial environments. Recently, research efforts aimed at filling the gap between full-fledged security solutions and UASNs requirements and challenges have been proposed in [2, 4]. Authors combine energy efficient security features based on both PKC and symmetric cryptography to provide message confidentiality, authentication, and secure routing protocols tailored to such communication constrained environment. However, authors assume that a key agreement protocol is already in place, or at least, that nodes are equipped with certificates and public keys. Key agreement protocols based on implicit certificates have not been explored yet on UASNs. However, a recent approach for Wireless Sensor Networks is presented by Galindo et al. in [10]. Their proposed schemes are based on the original MQV protocol [13] on elliptic curves (ECMQV), and supports both explicit certificates (e.g., public keys digitally signed by the sink) and implicit certificate (SC-ECMQV). However, as the proposed schemes only consider a two-way key agreement protocol, they lack perfect forward secrecy and are vulnerable to group representation and unknown key share attacks [11].

Authors in [15, 16] provide a formal proof of the vulnerabilities of MQV, and present the Fully Hashed MQV protocol (FHMQV), a new secure protocol based on both MQV and HMQV. FHMQV provides security, efficiency and resiliency against the ephemeral secret exponent leakage. In addition, FHMQV operations have been designed to be secure and lightweight in terms of computational overhead. For all the reasons, FHMQV can be reasonably implemented and used in UASNs.

3 THE KEY AGREEMENT PROTOCOL

In this section we describe the notation and the key agreement protocol. The proposed protocol is composed by two phases: The *start-up* and the *second* that are based on the ECQV and FHMQV schemes, respectively. Each node has two public keys named long-term and ephemeral. The long-term public keys are used to provide an additional mechanism for authentication. Instead, the ephemeral keys are needed for the Perfect Forward Secrecy (PFS). The PFS is a property that guarantee that session keys will not be affected even if the long-term private key is compromised [7]. Therefore ephemeral key must generated for each run of the protocol. In what follows we describe first our notation and then the two different phases performed by the key agreement protocol.

3.1 Notation

We use the *hat notation* for representing the node's identity. Hence $\hat{A}$ can be interpreted as A's identification code. We use upper case letters for the public keys and the lower case for the private ones. We denote $\overline{H}$ as a l-bit hash function where $l = (\lfloor log_2 q \rfloor + 1)/2$ [11]. In this context we use the multiplicative notation.

3.2 Start-up phase

We assume that a start-up phase is performed before the actual nodes deployment. During such a phase, the Certification Authority (CA) generates the implicit certificates and the long-term public keys for each node using the Elliptic Curve based Qu-Vanstone (ECQV) scheme. Then, the certificates and the long-term public keys are distributed to the network nodes. The long-term public key is bound to the entity for a defined period of time according to the certificates [13]. We assume that a sink node acts as CA (Certification Authority) and all the others are the peers that want to share the session keys.

In Protocol 1 we describe deeply the start-up phase:

Protocol 1 Start-up phase based on ECQV.

1: Let be defined E an elliptic curve over a finite field $\mathbb{Z}_p$, the cyclic subgroup as the set of points of this curve and the generator g of the cyclic subgroup of order n;
2: each node $\hat{A}$ of the network generates a point by selecting a random element $x \leftarrow_{\$} \mathbb{Z}_p$ and computing the point as $X = g^x$;
3: node $\hat{A}$ sends X to $\widehat{CA}$.
4: When $\widehat{CA}$ receives X, it does the following:
   a: selects a random element $q \leftarrow_{\$} \mathbb{Z}_p$;
   b: computes the points $E_Q = g^q$;
   c: computes $P = X \cdot E_Q$;
   d: generates the implicit certificate P_A appending to P the validity period;
   e: computes $sign := \overline{H}(P_A)q + key_{CA} \bmod n$, where key_{CA} is the private long term key of the $\widehat{CA}$.
   f: sends (P_A, $sign$) to $\hat{A}$.
5: When $\hat{A}$ receives (P_A, $sign$) it computes its private long term key $a := sign + \overline{H}(P_A) \cdot x \bmod n$.
6: $\hat{A}$ computes its long term public keys $A = P^{\overline{H}_{P_A}} E_{Q_{CA}}$, using the public key of the $\widehat{CA}$.

3.3 Second phase

In this section we briefly describe the second phase of the protocol based on the FHMQV[16] schema. All the steps of such protocol are shown in Protocol 2. Notice that the values a and b used in Protocol 2 are, respectively, the long term private keys of $\hat{A}$ and $\hat{B}$. The KDF is a key derivation function that produces an output of length 32 bytes.

Protocol 2 Second phase based on FHMQV.

1: $\hat{A}$ selects $e_A \leftarrow_{\$} \mathbb{Z}_p$ and computes the ephemeral exponent $X_A = g^{e_A}$;
2: $\hat{A}$ sends a message to $\hat{B}$ containing its implicit certificate P_A and X_A;
3: Once $\hat{B}$ receives the message, it performs the following:
   a: verifies P_A; if it is valid then $\hat{B}$ can retrieve A which is the long term public key of $\hat{A}$;
   b: selects $e_B \leftarrow_{\$} \mathbb{Z}_p$ and computes $Y_B = g^{e_B}$;
   c: $d := \overline{H}(X_A, Y_A, \hat{A}, \hat{B})$;
   d: $e := \overline{H}(Y_A, X_A, \hat{A}, \hat{B})$;
   e: $s_B := e_B + eb \bmod p$;
   f: $\sigma_B := (X_A A^d)^{s_B}$;
   g: computes $K_{MAC} = KDF(\sigma_B, \hat{A}, \hat{B}, X_A, Y_B)$;
   h: computes $MAC_{\hat{B}} = MAC_{K_{MAC}}(P_B, P_A, Y_B, X_A)$, where P_B is the implicit certificate of $\hat{B}$;
   i: sends a message composed by $(P_B, Y_B, MAC_{\hat{B}})$ to $\hat{A}$.
4: Once received (P_B, Y_B,$MAC_{\hat{B}}$), $\hat{A}$ performs the following:
   a: verifies P_B; if it is valid then $\hat{A}$ can retrieve B which is the long term public key of $\hat{B}$;
   b: $d := \overline{H}(X_A, Y_B, \hat{A}, \hat{B})$;
   c: $e := \overline{H}(Y_B, X_A, \hat{A}, \hat{B})$;
   d: $s_A := e_A + da \bmod p$;
   e: $\sigma_A := (Y_B B^e)^{s_A}$;
   f: computes $K_{MAC} = KDF(\sigma_A, \hat{A}, \hat{B}, X_A, Y_B)$;
   g: verifies if $MAC_{\hat{B}}$ is valid: If not, it aborts.
   h: computes $MAC_{\hat{A}} = MAC_{K_{MAC}}(P_A, P_B, X_A, Y_B)$ and sends $MAC_{\hat{A}}$ to $\hat{B}$;
   i: computes the shared key $K_{AB} = K_{MAC}$.
5: Once $\hat{B}$ receives $MAC_{\hat{A}}$, it verifies the $MAC_{\hat{A}}$ and then computes the shared key $K_{AB} = K_{MAC}$;
6: K_{AB} is the shared session key.

4 OPTIMIZATIONS AND IMPLEMENTATION DETAILS

In this section we briefly describe the functions used and some optimizations performed to increase the performance and, at the same time, to reduce the energy consumption of the proposed solution.

The entire protocol relies on SECGP recommended curve (SECGP-160) defined over a finite field $\mathbb{F}_p$ that guarantees an acceptable level of security of 80 bits. In this way, an attacker needs to perform 2^{80} operations to find a solution of the discrete logarithm problem, for a given instance, and therefore to obtain the private keys. We used the functions provided by the RELIC-toolkit (see [1] documentation for details) to perform operations over elliptic curve, selection of random elements and cryptographic hash functions. The MAC is implemented as hash-based authentication code using the function HMAC-SHA256. The KDF function is implemented using a SHA-256 cryptographic hash function, as well as for all the other hash functions. In the implemented protocol, the shared key is generated by performing only one iteration of SHA-256. This optimization shown in [3] allowed us to save time and slightly reduce the energy consumption without affecting the protocol's security.

In our protocol implementation we tried also to reduce as much as possible the size of the exchanged packets to keep at bay the energy consumption thus increasing the network lifetime. In particular, the size of the transmitted packets depends on both the size of the point exchanged and the MACs. This has led us to choose elliptic curve defined over the smallest finite field available, that provides a full-fledged

provable security, together with point compression algorithm. In particular, each point of the SECGP-160 curve can be represented by 40 bytes, that is 20B for each coordinate. This size can be reduced of about 50% by using the point compression algorithm, available in RELIC-toolkit. This allows us to transmit only the x-coordinate since each node already knows the elliptic curve used and therefore it can trivially calculate the y-coordinate. Therefore the resulting packet size for a single point will be of 21 bytes: 20B for the x-coordinates plus 1 byte used to speed-up the calculation at the receiver.

Similarly, we reduced the size of the MAC. The HMAC-SHA256, using a 256-bit long key, produces a MAC of 32 bytes long. We decided to truncate the tag of the MAC to 16 bytes without affecting the security [9].

These size optimizations allowed us to have a size of 58 bytes (42B and 16B for the points and the MAC, respectively) for the largest packet exchanged in the protocol.

5 SYSTEM ARCHITECTURE

In this section we describe the details of the experimental evaluation of the protocol. We start by describing the integration of this new protocol inside the SecFUN framework provided by the SUNSET Software Defined Communication Stack (SUNSET SDCS) [2, 14]. Then we describe the real system used to test and validate such protocol: The AppliCon SeaModem acoustic modem integrated with a BeagleBone Black embedded system.

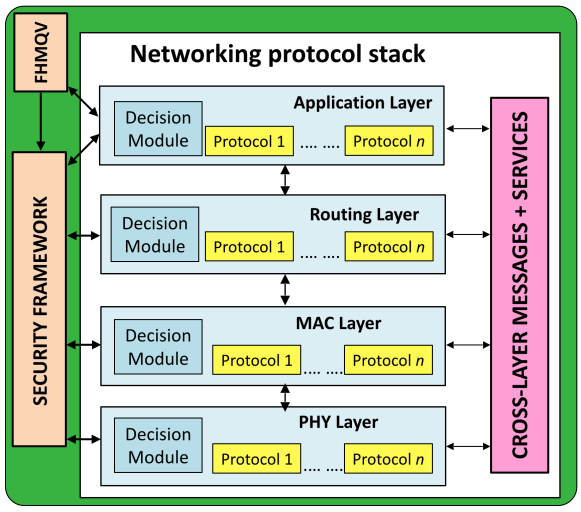

Figure 1: SUNSET SDCS Architecture.

5.1 SUNSET SDCS

The Sapienza University Networking framework for underwater Simulation Emulation and real-life Testing framework [14] (SUNSET) is a framework that provides networking and communication capabilities to underwater nodes. SUNSET supports concept of the Software Defined Communication Stack (SDCS) allowing to run different protocol stacks and modems dynamically and adaptively to the environmental conditions. It is currently commercialized by WSENSE srl. The proposed protocol has been implemented as a new module in the SUNSET SDCS framework as shown in Figure 1 by leveraging on the SecFUN framework. The module allows the user to authenticate and share a session key in order to enable encryption and authentication operations. This can be performed on-demand or in automatic way according to the specific application or application scenario. The protocols that are running at the application layer of the protocol stack can use our FHMQV implementation and once that the key agreement has been correctly executed, the session key can be used by SecFUN primitives.

5.2 Acoustic Modem

SeaModem [5] is a low cost MFSK underwater acoustic modem developed by AppliCon s.r.l. for shallow water communications, operating in the $25 - 40$ KHz frequency band. It offers several feature, such as different transmission data rate available of 750, 1500 and 2250 bits per second, error detection/correction algorithms and four different transmission power levels (from 5W up to 40W). In addition, it has a plugin connector that gives the capability to host a BeagleBone board. Since the BeagleBone is an open source embedded PC running a Linux OS, the developers can use the SeaModem as a Linux system. Since the BeagleBone is based on an ARM processor, our protocol code has been optimized to run efficiently and successfully on such system.

6 EXPERIMENTAL RESULTS

In this section we present the results related to the second phase of the FHMQV protocol (Figure 2) using real hardware. In particular, we assume that the generation and the distribution of the implicit certificates for each node are performed before the actual deployment in field. The proposed protocol has been tested using two SeaModems acoustic modems deployed in a water tank. The data packet payload size varies according to the packet type. In particular, the sizes of the (P_A and X_A), (P_B, Y_B, $MAC_{\hat{B}}$) and $MAC_{\hat{A}}$ packets are 42B, 58B, 16B, respectively. In addition, 17 bytes of header are added by SUNSET SDCS to each packet for the remaining layers of the protocol stack. The data rate of the modems has been set to 750 bits/s. We selected a guard period of 10s to reduce the impact of the multi-path effect at the receiver side but at a toll of reducing the data rate. The metrics we considered are: 1) the time spent to compute the messages to be transmitted and their actual transmission delay; 2) the energy consumed to compute, transmit and receive such messages. The BeagleBone board was running at 1 GHz resulting in an energy consumption of about 1.9W. The SeaModem energy consumption is instead of 5W and 0.5W for the transmission and reception mode, respectively.

The results are shown in Tables 1 and 2. It can be seen that the computational times required by the protocol operations are really small, even on an embedded system. The reason is

because the FHMQV protocol is highly efficient resulting in a small number of operations with a low computation cost. In addition the use of elliptic curves allows to further reduce the computational overhead, as shown in [18]. The energy consumption and the delay needed to actually transmit a packet in water are instead quite high. This is because the modem uses a high guard period that results in a low data rate and therefore in a higher energy consumption. For sake of clarity, we reported the effective transmission time including the serial transfer time and modem additional overheads.

Operations	Time	Energy
$\hat{A}$: X_A generation and P_A recovery	15.27 ms	0.03 J
$\hat{A}$: Verify $MAC_{\hat{B}}$, σ calculation, $MAC_{\hat{A}}$, key derivation	25.21 ms	0.05 J
$\hat{B}$: Y_B generation and P_B recovery	15.78 ms	0.03 J
$\hat{B}$: Verify $MAC_{\hat{A}}$, σ calculation, $MAC_{\hat{B}}$, key derivation	29.08 ms	0.06 J

Table 1: Computational times and energy consumption of the different operations performed both by node $\hat{B}$ and $\hat{A}$.

Message	Size (bytes)	Time TX	Energy TX	Energy RX
$\hat{A}$ sends (P_A and X_A)	42 + 17	7.6 s	38 J	3.80 J
$\hat{B}$ sends (P_B, Y_B, $MAC_{\hat{B}}$)	58 + 17	9.66 s	48.3 J	4.83 J
$\hat{A}$ sends $MAC_{\hat{A}}$	16 + 17	4.25 s	21.25 J	2.13 J

Table 2: Transmission delays and energy consumption of the FHMQV protocol.

7 CONCLUSIONS

In this paper we presented a key agreement protocol for UASNs based on FHMQV. We have developed and tested on real systems FHMQV based on Elliptic Curve Cryptography used with implicit certificates. The protocol makes possible that two honest parties share a common secret key starting from an authenticated public key. The proposed protocol has been implemented as a new module in the SUNSET SDCS framework by leveraging on the SecFUN framework. Several optimization tailored to UASNs have been performed to combine a full-fledged provable security with low overhead. The results confirm the feasibility of our implementation in terms of computational time and energy consumption.

8 ACKNOWLEDGMENTS

This work has been partially supported by the EC-EASME ARCHEOSUb (Autonomous underwater Robotic and sensing systems for Cultural HEritage discovery cOnServation and in sitU valorization) Project.

REFERENCES

[1] D. F. Aranha and C. P. L. Gouvêa. [n. d.]. RELIC is an Efficient LIbrary for Cryptography. ([n. d.]). Retrieved October 06, 2017 from https://github.com/relic-toolkit/relic

[2] G. Ateniese, A. Capossele, P. Gjanci, C. Petrioli, and D. Spaccini. 2015. SecFUN: Security framework for underwater acoustic sensor networks. In *Proceedings of MTS/IEEE OCEANS 2015*. Genova, Italy, 1–9. https://doi.org/10.1109/OCEANS-Genova.2015.7271735

[3] A. Capossele, V. Cervo, C. Petrioli, and D. Spenza. 2016. Counteracting Denial-of-Sleep Attacks in Wake-up-radio-based Sensing Systems. In *Proceedings of the 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 1–9.

[4] A. Capossele, G. De Cicco, and C. Petrioli. 2015. R-CARP: A Reputation Based Channel Aware Routing Protocol for Underwater Acoustic Sensor Networks. In *Proceedings of the 10th International Conference on Underwater Networks & Systems (WUWNET '15)*. ACM, New York, NY, USA, Article 37, 6 pages. https://doi.org/10.1145/2831296.2831339

[5] G. Cario, A. Casavola, M. Lupia, and C. Rosace. 2015. SeaModem: A low-cost underwater acoustic modem for shallow water communication. In *Proceedings of MTS/IEEE OCEANS 2015*. Genova, Italy, 1–6. https://doi.org/10.1109/OCEANS-Genova.2015.7271721

[6] G. Cong, Y.and Yang, Z. Wei, and W. Zhou. 2010. Security in underwater sensor network. In *Proceedings of the International Conference on Communications and Mobile Computing (CMC)*, Vol. 1. IEEE, 162–168.

[7] C. Cremers and M. Feltz. 2015. Beyond eCK: perfect forward secrecy under actor compromise and ephemeral-key reveal. *Designs, Codes and Cryptography* 74, 1 (2015), 183–218.

[8] M. C. Domingo. 2011. Securing Underwater Wireless Communication Networks. *Wireless Communications* 18, 1 (feb 2011), 22–28.

[9] M. J. Dworkin. 2007. *NIST SP 800-38D. Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*. Technical Report. Gaithersburg, MD, United States.

[10] D. Galindo, R. Roman, and J. Lopez. 2012. On the Energy Cost of Authenticated Key Agreement in Wireless Sensor Networks. *Wireless Communications and Mobile Computing* 12 (Jan 2012 2012), 133–143. https://doi.org/10.1002/wcm.894

[11] H. Krawczyk. 2005. HMQV: A High-Performance Secure Diffie-Hellman Protocol (Extended Abstract). In *Proceedings of CRYPTO 2005*. Springer.

[12] C. Lal, R. Petroccia, M. Conti, and J. Alves. 2016. Secure underwater acoustic networks: Current and future research directions. In *Proceedings of 2016 IEEE Third Underwater Communications and Networking Conference (UComms)*. 1–5. https://doi.org/10.1109/UComms.2016.7583466

[13] L. Law, A. Menezes, M. Qu, J. Solinas, and S. Vanstone. 1999. An efficient protocol for authenticated key agreement. *Designs, Codes and Cryptography* (1999).

[14] C. Petrioli, R. Petroccia, J. R. Potter, and D. Spaccini. 2015. The SUNSET framework for simulation, emulation and at-sea testing of underwater wireless sensor networks. *Ad Hoc Networks* 34 (2015), 224–238. https://doi.org/10.1016/j.adhoc.2014.08.012

[15] A. P. Sarr and V. P. Elbaz. 2016. On the Security of the (F) HMQV Protocol. In *Proceedings of International Conference on Cryptology in Africa*. Springer, 207–224.

[16] A. P. Sarr, V. P. Elbaz, and J. C. Bajard. 2009. A Secure and Efficient Authenticated Diffie-Hellman Protocol. In *Proceedings of EuroPKI*. Springer, 83–98.

[17] S. Sciancalepore, A. Capossele, G. Piro, G. Boggia, and G. Bianchi. 2015. Key Management Protocol with Implicit Certificates for IoT Systems. In *Proceedings of the 2015 Workshop on IoT Challenges in Mobile and Industrial Systems (IoT-Sys '15)*. ACM, New York, NY, USA, 37–42. https://doi.org/10.1145/2753476.2753477

[18] E. Souza, H.C. Wong, I. Cunha, A.A.F. Loureiro, L.F.M. Vieira, and L.B. Oliveira. 2013. End-to-end authentication in Under-Water Sensor Networks. In *Proceedings of the 18th IEEE International Symposium on Computers and Communications (ISCC 2013)*. Split, Croazia, 000299–000304. https://doi.org/10.1109/ISCC.2013.6754963

[19] H. Yan, Z. J. Shi, and Y. Fei. 2009. Efficient implementation of elliptic curve cryptography on DSP for underwater sensor networks. In *Proceedings of the 7th Workshop on Optimizations for DSP and Embedded Systems (ODES-7)*. 7–15.