SecFUN: Security Framework for Underwater acoustic sensor Networks

Giuseppe Ateniese,* Angelo Capossele,*† Petrika Gjanci,*† Chiara Petrioli*† and Daniele Spaccini*†
* Dipartimento di Informatica
Università di Roma "La Sapienza," Rome, Italy
{ateniese, capossele, gjanci, petrioli, spaccini}@di.uniroma1.it
† WSENSE s.r.l., Rome, Italy

***Abstract*—In this paper we introduce SecFUN, a security framework for underwater acoustic sensor networks (UASNs). Despite the increasing interest on UASNs, solutions to secure protocols from the network layer up to the application layer are still overlooked. The aim of this work is therefore manyfold. We first discuss common threats and countermeasures for UASNs. Then, we select the most effective cryptographic primitives to build our security framework (SecFUN). We show that SecFUN is flexible and configurable with different features and security levels to satisfy UASN deployment security requirements. SecFUN provides data confidentiality, integrity, authentication and non-repudiation by exploiting as building blocks AES in the Galois Counter Mode (GCM) and short digital signature algorithms. As a proof of concept of the proposed approach, we extend the implementation of the Channel-Aware Routing Protocol (CARP) to support the proposed cryptographic primitives. Finally, we run a performance evaluation of our proposed secure version of CARP in terms of the overall energy consumption and latency, employing GCM and the state of the art in short digital signature schemes such as ZSS, BLS and Quartz. Results show that a flexible and full-fledged security solution tailored to meet the requirements of UASNs can be provided at reasonable costs.**

***Index Terms*—Underwater security, underwater sensor networks, underwater protocols, CARP, SUNSET.**

I. Introduction

The interest of both academia and industry on Underwater Acoustic Sensor Networks (UASNs) has been steadily increasing in recent years. UASNs are becoming the key enabler for a large set of application scenarios ranging from scientific exploration and commercial exploitation, to homeland security [1]. Novel communication protocols and cooperative coordination algorithms [2] have been proposed in the literature to enable collaborative monitoring tasks performed by teams of heterogeneous static and mobile underwater platforms. However, such solutions fail to consider security as a key performance indicator. Spoofing, altering, or replaying routing information can affect the entire network, making UASNs vulnerable to routing attacks such as selective forwarding, sinkhole attack, Sybil attack, HELLO flood attack and acknowledgment spoofing [3]. The lack of security support is startling if we observe that security is indeed an important requirement in many emerging civilian and military applications, such as pipeline and environmental monitoring, strategic surveillance and reconnaissance, etc. In particular, some deployments require the establishment of a secure channel from sensor nodes to an infrastructure sink to provide confidentiality. Other applications require message authentication and integrity via digital signatures. Although attacks against UASNs are similar to the ones against terrestrial Wireless Sensor Networks (WSNs) and Mobile Ad-hoc Networks (MANETs), the same countermeasures are not directly applicable to UASNs due to the different characteristics of such networks. In particular, UASNs nodes communicate using acoustic waves, experiencing lower bandwidth and bit rate, higher propagation delays, and higher energy consumption than those of WSNs. The computational power available on underwater platforms is instead much higher than that of typical WSNs. As a result, a heavy computation performed in UASNs may have a lower impact on energy consumption and cause fewer network delays than sending a large message [4]. This in turn poses the important research question of how to design a complete security framework that minimizes the overhead (and associated energy consumption) due to the extra information that needs to be transmitted, while ensuring the highest security levels.

In this paper we analyze the most effective cryptographic primitives suitable for UASNs, and we detail how they can be applied to secure protocols from the network layer up to the application layer. We show that the cost of adopting full-fledged security solutions can be minimal in terms of performance and energy consumption. Our approach can be used to extend both flooding-based and unicast routing protocols, to provide confidentiality, integrity, and authentication. Our security framework, SecFUN (for Security Framework for Underwater acoustic sensor Networks), exploits as building block the Galois Counter Mode (GCM) [5], which is a mode of operation to encrypt and authenticate data using a 128-bit block cipher, such as AES. GCM presents several features well suited for UASNs. First of all, the length of the ciphertext is equal to the length of the plaintext, thus GCM does not introduce overhead (other than a small tag used for authentication whose length varies from 0 to 128 bits). Furthermore, the integrity and the authentication of the encrypted data is provided by a tag which can be verified without performing any decryption operations (which is essential to thwart denial-of-service attacks, etc.). Finally, GCM provides the authenticated-only message mode, called GMAC, that is useful when encryption is not required. GCM is based on the counter mode thus it is fully parallelizable and

employs only the encryption algorithm for both encryption and decryption (crucial when AES is used). Another more subtle advantage of GCM is that it can act as an incremental MAC, thus making it possible to recompute authentication tags on dynamic data by only accessing the data portions that have changed. As a proof of concept of the proposed approach, we extend the implementation of CARP [6] to support security. CARP, for Channel-aware Routing Protocol, is a cross-layer routing protocol that exploits link quality information for data forwarding and is designed to be robust, energy-aware and adaptive. We enhanced the basic version of CARP to support the AES-GCM encryption, named S_e-CARP, by assuming that each node of the network shares the same group key and a unique secret key with the sink. This key is then used by the nodes to encrypt and authenticate all the packets exchanged in the network. Specifically, nodes are able to encrypt data, such as sensed data from their sensors, at the application layer, using the key shared with the sink. Our security framework also supports short digital signatures, such as BLS [7], ZSS [8] and Quartz [9], to provide source authentication with non-repudiation at the application layer. To this aim, we have further extended CARP, termed S_{ds}-CARP, to support source/message authentication through such digital signature schemes. The communication overhead can be further improved in S_{ds}-CARP through the support of BLS signature aggregation, where multiple BLS signatures are accumulated into a single value. We assess the performance of our extended versions of CARP by using the SUNSET framework [10] with underwater monitoring networks composed of 20 nodes. Bellhop [11] is used to compute acoustic path loss at a given location, as well as the spatially-varying interference induced by node transmissions. The comparison between CARP, S_e-CARP and S_{ds}-CARP are reported in terms of energy consumption and latency. Results show that when transmitting large data packets, the energy consumption in S_e-CARP increases by a value between 20% and 53%. Latency in S_e-CARP also increases by a value between 12% and 55%. Similarly, using different digital signatures in S_{ds}-CARP leads to different energy consumption of the network according to the overhead introduced by each scheme. In particular, the energy consumption of Quartz, ZSS and BLS schemes is, respectively, 61% (6%), 31% (3%) and 29% (1.5%) greater than that of CARP when transmitting short (large) data packet. The lowest difference in the performance is obtained when the signatures aggregation feature of BLS is exploited, saving up to 19% and 37% of the overall energy consumed by ZSS and Quartz, respectively.

The rest of the paper is organized as follows. Previous work on security in UASNs and related attacks are summarized in Section II and Section III. In Section IV we define the secure primitives implemented in the proposed framework. A proof of concept of our approach is reported in Section V and related results are shown in Section VI. Finally, Section VII concludes the paper.

II. Related work

Acknowledging the importance of UASN security, several works [12], [13] have recently discussed security issues in UASNs, underlining the peculiarities of these networks and analyzing threats, attacks, and possible countermeasures. Several authors pointed out the need for the research community to focus on the design of new security protocols but with an emphasis on properties such as energy efficiency and the ability to adapt to the requirements imposed by the specific application/scenario. More recently, some works have investigated novel security solutions tailored to the unique features of UASNs. Among the first attempts to improve security in underwater networks, the authors in [14] proposed the implementation of Elliptic Curve Cryptography (ECC), exploiting the Digital Signal Processors in existing acoustic modems to speed up the computation of cryptographic operations. Their results show that an efficient ECC implementation can reduce the computational overhead. However, their work does not analyze the communication overhead incurred when ECC is adopted. Security protocols, in addition, require a mechanism to exchange secret keys between peers. The authors in [15] show that, in extremely constrained environments such as UASNs, the most effective solution to the problem of key distribution is to employ a non-interactive identity-based key establishment protocol such as SOK [16]. These schemes are based on bilinear maps and in theory require about 384 bits to let peers exchange secret keys. Their security, however, must account for recent quasi-polynomial attacks on the discrete logarithm problem on small characteristic [17]. Souza et al. address the problem of message authentication in [4]. The authors compare the energy efficiency of different digital signature algorithms in both underwater and terrestrial scenarios but without integrating such algorithms into network protocols and evaluating them in a typical UASNs scenario. When dealing with UASNs, finding the tradeoff between security and energy efficiency can be difficult. Dini et al. [18] propose a solution to secure both unicast and multicast communications in underwater acoustic sensor networks. Their solution provides confidentiality and message integrity and introduces little overhead in the network, but at the cost of a weaker security level. All papers discussed so far consider cryptographic aspects to security in UASNs. Other mechanisms are needed to prevent attacks such as the wormhole attack. The work [19] provides a solution for UASNs based on the Direction of Arrival (DoA) estimation to protect neighbor discovery protocols from the wormhole attack. Their solution does not require secure and accurate time synchronization or localization, but can be affected by orientation error among sensors. A similar approach is presented in [20] where each node collects the distance estimations from its neighbors to reconstruct the local network topology. In this way, nodes can detect wormhole attacks in a distributed manner. However, other attacks can still be concealed by manipulating the buffering times of distance estimation packets.

Attack Name	Network stack layer(s)
Sybil Attack	Application, Routing, MAC
Hello Flood Attack	Routing, MAC
Acknowledgment Spoofing	MAC
Replay Attack	Application, Routing, MAC
Exhaustion	Application, Routing, MAC
Selective forwarding attack	Routing
Sinkhole attack	Routing

Table I: Attack types and related network stack layers.

III. Attack types

The broadcasting nature of the UASN channel makes the data vulnerable to being modified, injected and eavesdropped. The injection in the network of fake data such as spoofed, replayed or altered information could disrupt the regular network flow, creating loops, attracting or rejecting traffic in specific areas, partitioning the network or creating bad routes [12], [13]. Different attacks, described in what follows, can affect different layers of the protocol stack as shown in Table I:

- *Sybil Attack:* The aim of an attacker is to forge multiple identities in the network in order to appear in multiple locations at once;
- *Hello Flood Attack:* Malicious nodes broadcast a hello message using a long range transmission pretending to be a neighbor of the nodes receiving such a message;
- *Acknowledgment Spoofing*: If a protocol uses link-layer acknowledgments, a malicious node can generate false acknowledgments to broadcast false information about network links;
- *Replay Attack*: A valid data transmission is replied or delayed;
- *Exhaustion*: A malicious node can reduce the network lifetime by forcing network nodes to process useless messages;
- *Selective forwarding attack*: A malicious node can act as a relay for several nodes deciding selectively which packets may be either forwarded or dropped;
- *Sinkhole attack*: The attacker increases the probability to be chosen as a relay by advertising zero-cost routes to every other node.

It is therefore clear that a cross-layer security approach is needed to face with these heterogeneous attacks. In order to prevent these threats, the security protocols need to provide at least:

- Confidentiality: Data must be protected against unauthorized read through data encryption. It should be robust against nodes being compromised: compromising a single node or a few nodes should not compromise the security of the entire network [21].
- Integrity: It should guarantee that the received messages are not altered in transit by the attackers. A keyed cryptographic tag, such as a Message Authentication Code (MAC), can protect packets against modification [21].
- Availability: It indicates the capability to provide services whenever they are required. The most widespread threat to network availability is a denial of service (DoS) attack [12], [13]. This happens when attackers generate interferences or decrease the power of nodes through various methods such as the exhaustion attack [21].
- Freshness: It could refer to data freshness or key freshness. Data freshness suggests that data is recent, and it ensures that no old messages have been replayed. Key freshness typically ensures that shared keys are changed over time to prevent a replay attack. A nonce, or other time-related counters, can be added into the packet to ensure data freshness and a re-keying process can be performed to ensure key freshness.
- Authentication: It prevents false data injection and verifies user identities. Digital signatures or simply MACs can be used to authenticate the origin of a message [21].
- Non-repudiation: It ensures that a node cannot deny having sent a message [21]. Digital signatures provide this property.

IV. SecFUN primitives

Cryptographic primitives are used to implement several security services. The algorithms based on cryptography can be divided in two families: Symmetric (or secret-key) and asymmetric (or public-key) cryptography. The first one uses the same key for encryption and decryption, while the latter uses two different keys. Asymmetric-key cryptography (e.g., the RSA algorithm) requires more computational resources than symmetric-key cryptography (e.g., the AES block cipher). Thus, hybrid schemes are usually adopted: public-key schemes for key exchange and non-repudiation and secret-key schemes for MAC and data confidentiality. Our proposed framework, SecFUN, provides both symmetric and asymmetric based cryptography to support message authentication, replay protection, and confidentiality along with a flexible selection of MAC sizes, and message/entity authentication and integrity via digital signatures. Specifically, it provides a cross-layer protection from the link-layer up to the application layer by a configurable and flexible selection of security features that can be tailored to the needs of the specific application/scenario. The cryptographic primitives in SecFUN are briefly reviewed next.

A. Symmetric-key based encryption and authentication

To guarantee confidentiality, authentication and integrity of critical messages (such as routing information), we selected the Galois Counter Mode [5] (GCM) as the SecFUN block cipher mode of operation. GCM can provide support for: 1) message confidentiality through encryption; 2) authentication-only message mode, termed GMAC, that can be used when encryption is not required; 3) both authenticated and encrypted messages. The flexibility of GCM makes it an ideal choice for underwater communication. In addition, GCM has been designed to natively support message authentication in "one pass". As a result, it is more efficient than other modes of operation, such as the Cipher Block Chaining Mode (CBC) that requires additional message integrity check algorithms

(e.g., CBC-MAC, HMAC), thus increasing the overall overhead. Finally, the ciphertext produced by GCM has the same length of the original plaintext which is ideal for devices with bandwidth constraints, such as UASNs.

SecFUN's version of GCM uses a 128-bit cipher (AES) for encryption. Message authentication and integrity is provided by computing the Message Authentication Code (MAC) in the Galois field. The inputs of GCM are a symmetric key *K*, an initialization vector *IV*, a plaintext *P*, and any additional data *A* to be authenticated. The output consists of the ciphertext *C* and an authentication tag *T*, which is used to verify both the integrity and the authenticity of the encrypted data. Such a tag can be verified without performing decryption operations thus preventing exhaustion attacks (e.g., a malicious node trying to deplete a node's energy charge by forcing it to process bogus messages). The length of *T* can be any value between *0* and *128*, according to the specific security level required. In particular, NIST [5] recommends a size ranging from 64 to 128 bits.

B. Asymmetric-key based authentication: Digital Signatures

Using short messages is of paramount importance in UASNs to reduce the very high energy consumption related to their acoustic transmissions. For this reason, the use of the most popular asymmetric primitives, such as RSA, is not recommended since keys and signatures are very large. In recent years, new cryptographic schemes have been devised that are more suitable for resource and bandwidth constrained devices [22], [23], [24], [25], as in the case of UASNs. An example is Elliptic Curve Cryptography (ECC). A cryptosystem based on elliptic curves is defined by a finite field F_q, a small set of parameters that describe the elliptic curve E/F_q, a point $P \in E(F_q)$, and the order n of P. These parameters are chosen in such a way that the elliptic curve discrete logarithm problem (ECDLP) cannot be solved by an efficient adversary in reasonable time. ECC promises to deliver the same security of RSA with much shorter keys and signatures. Within ECC, we considered schemes that require a bilinear map (pairing-based cryptography). In particular, we selected the Boneh-Lynn-Shacham [7] (BLS) and the Zhang-Safavi-Naini-Susilo schemes [8] (ZSS). Both schemes belong to the family of *short signatures*: The size of the signature is about 160 bits with a security level of 2^{80}. The BLS scheme supports, in addition, signature aggregation, i.e., signatures from different signers and on distinct messages can be accumulated into a single short value. In addition, the generation of actual signatures in both BLS and ZSS scheme is computationally efficient. Verification instead requires more effort but batch verification techniques can be adopted to mitigate the issue.

Signatures shorter than 160 bits have also been proposed. These schemes belong to multivariate cryptography [9] (e.g., HFE, Balanced Oil & Vinegar), where the underlying problem, known as *MinRank*, is based on the difficulty of solving systems of quadratic polynomial equations for sufficiently many quadratic unknowns $x_1, \ldots, x_n$.[1] Moreover, these schemes have typically two independent security parameters: The extension degree h and the degree of the hidden polynomial d. This makes them much more flexible than the ECC-based schemes in that the first parameter can be small to achieve shorter signatures, and the other one can be independently tuned to achieve the desired security level. Among these schemes, we selected Quartz [9] that provides signatures of 128 bits with a security level of 2^{80}. However, signature generation in Quartz is quite expensive since it involves the computation of (1) the four roots of the private polynomial P with degree d, and (2) the multiplication of two affine polynomials.

V. Proof of concept: S-CARP protocol families

In this section we describe the families of the secure version of CARP [6], termed S-CARP, that support the security services provided by the proposed framework. We denote with S_e-CARP and S_{ds}-CARP, the family of S-CARP that supports encryption primitives and digital signature schemes, respectively. In particular, the family of S_e-CARP is composed of S_e^8-CARP, S_e^{12}-CARP and S_e^{16}-CARP protocols that implement different security levels of 8B, 12B and 16B, respectively, as recommended by NIST [5]. The family S_{ds}-CARP is composed of S_{ds}^{Q}-CARP, S_{ds}^{ZSS}-CARP and S_{ds}^{BLS}-CARP protocols that implement Quartz, ZSS and BLS, respectively.

A. CARP

CARP [6] is a cross-layer routing protocol that exploits link quality information for data forwarding and is designed to be robust, energy-aware, and adaptive. When the protocol starts, a set-up phase is performed by broadcasting `HELLO` packets thus allowing the network nodes to acquire hop distance information from the sink. When a node x has one or more data packets to forward, it broadcasts a request message (`PING`) to choose the best suitable relay among its neighbors. Nodes receiving the transmitted request reply with a `PONG` response message. Each `PONG` message, sent by a neighbor y, contains information on y including: 1) Estimated hop distance from the sink; 2) Available buffer space; 3) Residual energy; 4) Estimated quality of the link between x and y and of the best link among those from y to its neighbors z. This information allows x to select the most suitable relay y among its neighbors. Once the node y successfully receives the data packet from x, it replies with an `ACK` packet. Optimization is performed by sending (1) multiple packets at a time and (2) cumulative acknowledgments to reduce the overhead introduced by the handshaking phase.

B. S_e-CARP

In the encryption-enabled version of CARP, each node shares the same group key and a unique secret key with the

[1] Solving systems of multivariate quadratic polynomial equations under this assumption is proven to be NP-Hard or NP-Complete. Therefore schemes based on multivariate cryptography are considered to be good candidates for post-quantum cryptography.

sink. When a node x has to forward a data packet to the sink, it broadcasts an encrypted and authenticated request packet (PING) using the group key, to find the best relay among its neighbors. All the nodes receiving a PING packet, after verifying the authenticity and integrity of the received PING, reply with an encrypted and authenticated response packets (PONG) that contains all the information needed by node x to choose the next hop relay y. Similarly, the packet data transmitted by x and the related ACK packet sent by y are encrypted and authenticated. [2]

VI. Performance Evaluation

This section describes the comparative performance of CARP with the protocols of the secure families S_e-CARP and S_{ds}-CARP. All the protocols have been implemented in SUNSET [10] on top of ns-2 [26], connected to the Bellhop ray tracing tool [11] via the WOSS interface [27]. Bellhop is used to compute acoustic path loss at a given location, as well as the spatially-varying interference induced by node transmissions. The historical environmental data input to Bellhop refer to an area located in the Norwegian fjord off the coast of Trondheim, with the coordinate $(0, 0, 0)$ of the surface located at $63°, 29', 1.0752''N$ and $10°, 32', 46.6728''E$. Sound speed profiles (SSP), bathymetry profiles and information on the type of bottom sediments of the selected area are obtained from the World Ocean Database [28], from the General Bathymetric Chart of the Oceans (GEBCO) [29] and from the National Geophysical Data Center Deck41 data-base [30], respectively. In the following we first describe the selected scenarios and protocol parameters settings (Section VI-A), we then discuss the metrics that we have investigated (Section VI-A) and we finally report on the results of our simulation experiments according to the different secure primitive implementations (Section VI-B and Section VI-B).

A. Simulation scenarios and settings

We consider a static UASN with 20 nodes (19 nodes plus the sink) randomly and uniformly placed in a region with surface of 2 km $\times$ 1 km at different depths, ranging from 10 to 240m. The sink is located at a side of the deployment area, on the surface. We simulate a scenario where random events occur in different zones of the network. Each zone is composed of a fixed number of nodes that start generating traffic when an event is detected according to a fixed sample rate of λ packets per second. In particular, λ takes values in the set $\{0.006, 0.01, 0.02, 0.033, 0.066, 0.1\}$. Since we assume that the same number of packets is generated for each event, the duration of an event depends on the traffic load: When the traffic load is higher, the event duration will be shorter. The destination of all packets is the sink. The average number of hops from source nodes to the sink is 2.3, while the maximum number of hops is 4. The data packet payload size (in bytes) varies in the set $\{20, 200, 600\}$ to simulate different application scenarios. The total size of a data packet is given by the selected payloads plus the headers added by the different layers. The physical header overhead changes according to the data rate but it is dominated by a 10ms synchronization preamble. The medium access control headers of all the versions of CARP are 4B long. The size of PING and PONG packets is 11B and 7B, respectively; the ACK packets are 6B long, while HELLO packets are 6B long. When using S_e-CARP, we encrypt each control and data packet by adding an additional payload of 8B, 12B and 16B according to the considered security level. Instead, S^Q_{ds}-CARP, S^{ZSS}_{ds}-CARP and S^{BLS}_{ds}-CARP add to each data packet a digital signature having size of 16B, 20B and 20B, respectively. In our simulations, we assume BPSK modulation. The carrier frequency is 25.6kHz for a bandwidth of 4000Hz. Bandwidth efficiency is set to 1bps/Hz, resulting in a data rate R_b that is equal to 1000b/s. The transmission power for short control packets (HELLO, PING, PONG and ACK) and data packets is set to 3.3W and 8W, respectively. The reception power consumption is set to 0.5W. Reception and transmission powers are estimated based on the energy consumption of existing acoustic modems. In our simulations we assume that the nodes share the same group key and a unique secret key with the sink. In addition, the sink knows the public keys of all the network nodes to verify all the digitally-signed messages.

Simulation metrics: The overhead introduced by authenticated encryption with different security levels and signature schemes on delivering data to the sink is assessed through the analysis of the following metrics.

- *End-to-end latency*, defined as the time between the packet generation and the time of its correct delivery at the sink.
- *Energy per bit*, defined as the energy consumed by the network to correctly deliver a bit of data to the sink.

The packet delivery ratio (PDR) at the sink, defined as the ratio between the packets correctly received by the sink and the packets generated by the nodes, is always greater than 95% in all the considered scenarios.

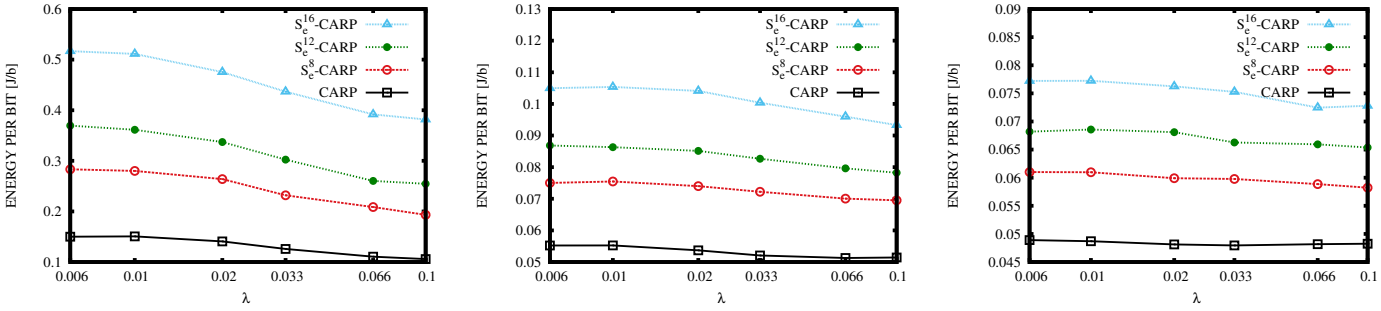
Scheme	Computational time	Energy consumption	Signature size
Quartz	7 sec.	9.24 J.	16B
BLS	0.025 sec.	0.34 J.	20B
ZSS	0.007 sec.	0.01 J.	20B

Table II: Computational times, power consumptions and sizes of different digital signature schemes.

Computational energy consumption: In Table II we show the trade-off between the energy consumption related to the computational time and the size of the digital signature for each considered scheme.

We evaluated the average energy consumption by means of the formula $E = U \cdot I \cdot t$, where t is the time to perform an operation, U is the voltage and I is the current intensity. The time t has been experimentally evaluated by performing tests which execute the selected digital signature operations 10.000 times, and recording the time needed to perform the overall cycle. This allows us to estimate the average time spent to

[2]The HELLO packets exchanged during the set-up phase are encrypted and authenticated as well.



(a) 20 Bytes. (b) 200 Bytes. (c) 600 Bytes.

Figure 1: S_e-CARP - Energy per bit with different security levels.

perform each signature. In order to validate our framework, we have tested the digital signature schemes on a typical underwater embedded device, specifically the Gumstix verdex pro platform (CPU: Marvell PXA270@600MHz, RAM: 128MB). The values U and I are derived from the data-sheet at [31].

B. Simulation results

Authenticated Encryption

In this section we compare CARP with S_e-CARP that implements encryption and authentication via MAC according to the different security levels described in Section IV-A. Therefore, in S_e^8-CARP, S_e^{12}-CARP and S_e^{16}-CARP both control and data packets transmitted are encrypted and authenticated resulting in an overhead of 8B, 12B and 16B, respectively.

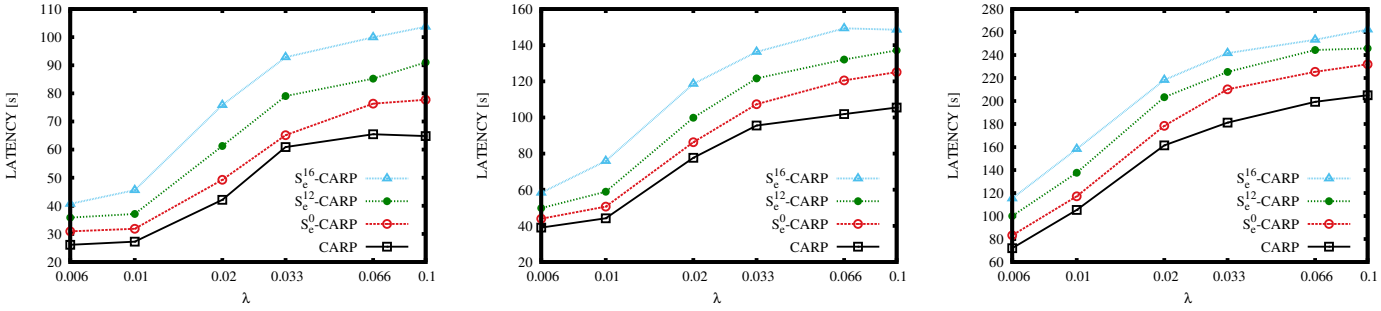
Energy per bit. Figure 1 shows the average energy per bit consumption of S_e-CARP considering different security levels for increasing traffic λ and for three different packet payloads. As expected, for higher security level the energy per bit consumed to deliver a bit of data increases with respect to CARP. When transmitting very short data packets, as shown in Figure 1a, the total energy per bit consumption increases significantly with the security level adopted. This is because the overhead introduced by the security level is comparable to the size of control and data packets resulting in a significant increment of transmission time and therefore energy per bit consumption. Therefore, the energy consumed by CARP is up to 47%, 59% and 70% less than that of S_e^8-CARP, S_e^{12}-CARP and S_e^{16}-CARP, respectively. As the packet size increases the impact of the authenticated encryption overhead decreases with respect to the size of the data packet payloads. In particular, the energy consumed by CARP is 18% (26%), 26% (36%) and 35% (47%) less than that of S_e^8-CARP, S_e^{12}-CARP and S_e^{16}-CARP, respectively, when considering data packets of 600B (200B). These results are shown in Figure 1b and Figure 1c. As both the traffic load and packet size increase, the energy per bit decreases in all the considered scenarios. This is because at higher traffic load the data packets are buffered at the nodes and then transmitted using trains of packet, thus maintaining high the packet delivery ratio and, at the same time, reducing the number of control packets exchanged and the amount of energy per bit spent. When the packet size increases (e.g., 600B), all the protocols consume less energy per bit (Figure 1c) since the number of bits delivered to the sink is higher.

End-to-end latency. The average end-to-end latency experienced by data packets successfully delivered to the sink is shown in Figure 2. Increasing the security level results in higher latency due to the overhead introduced by the encryption scheme in both control and data packets that leads to a higher number of packet collisions and retransmissions. When considering a small packet size, Figure 2a, the latency experienced by S_e^8-CARP, S_e^{12}-CARP and S_e^{16}-CARP increases of about 18%, 37% and 55%, respectively, with respect to CARP. As the packet size increases, the end-to-end latency increases due to larger transmission times, as shown in Figure 2b and Figure 2c. In particular, when the packet size is 600B (200B), S_e^8-CARP, S_e^{12}-CARP and S_e^{16}-CARP deliver the data packets with a latency increment of 12% (12%), 32% (27%) and 55% (49%), respectively, with respect to CARP. As the traffic rate increases, the end-to-end latency increases with a factor of 160% on average for each encryption scheme in the three packet sizes.

Digital signatures

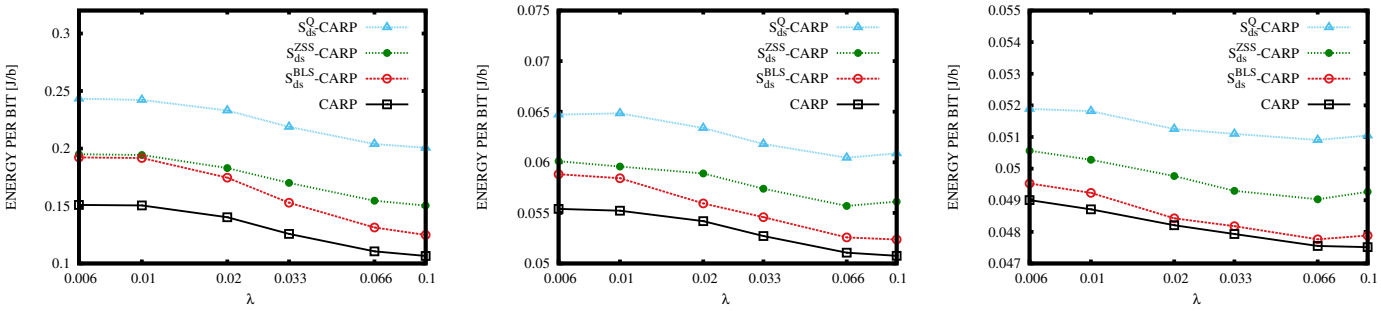
In this section we compare the performance of CARP with those of S_{ds}^Q-CARP, S_{ds}^{BLS}-CARP and S_{ds}^{ZSS}-CARP that implement different digital signature schemes (Quartz, BLS and ZSS, respectively).

Energy per bit. Figure 3 reports the average energy per bit consumed by the four protocols while performing digital signatures considering three different packets sizes. The results in Figure 3 show that the energy per bit consumption of S_{ds}^Q-CARP is always the highest with respect to the others protocols for all the considered packet data sizes and traffic loads. This is because the very high computational cost of signing a message with Quartz significantly affects the overall energy per bit consumption of S_{ds}^Q-CARP, even if the overhead introduced by its digital signature is 4B smaller than those of S_{ds}^{BLS}-CARP and S_{ds}^{ZSS}-CARP. Therefore when considering the packet size of 20B (Figure 3a), the energy per bit consumed by S_{ds}^Q-CARP is 61% higher than that of CARP while S_{ds}^{ZSS}-CARP and S_{ds}^{BLS}-CARP consume only 31% and



(a) 20 Bytes (b) 200 Bytes (c) 600 Bytes

Figure 2: S_e-CARP - End-to-end latency with different security levels.



(a) 20 Bytes (b) 200 Bytes (c) 600 Bytes

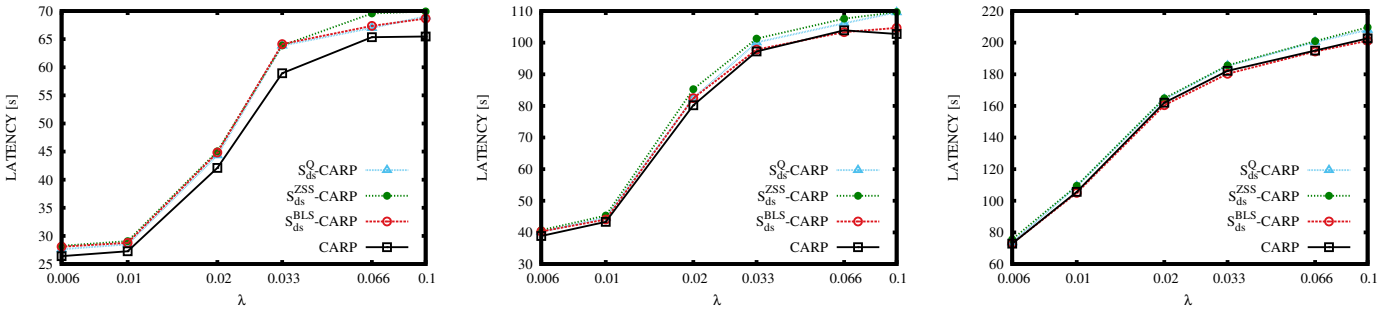
Figure 3: S_{ds}-CARP - Energy per bit with different digital signatures schemes.

29% more than CARP. When the packet size increases to 600B (200B), the energy per bit consumed by S_{ds}^{Q}-CARP, S_{ds}^{ZSS}-CARP and S_{ds}^{BLS}-CARP is 6% (16%), 3% (9%) and 1.5% (6%), respectively, higher than that spent by CARP. As both the traffic load and the packet size increase the energy saved by S_{ds}^{BLS}-CARP increases thanks to the capability of the BLS scheme to aggregate multiple data packets using a single digital signature. Therefore, when considering low traffic load (i.e., $\lambda = 0.006$) and short packet size (i.e., 20B), the energy consumed by S_{ds}^{BLS}-CARP and S_{ds}^{ZSS}-CARP is quite the same since the number of packet trains sent is low. When the number and size of packet trains increase due to higher traffic loads and (or) bigger packet data sizes, the energy saved by S_{ds}^{BLS}-CARP with respect to S_{ds}^{ZSS}-CARP increases as well. For instance, when $\lambda = 0.1$, S_{ds}^{BLS}-CARP saves up to 19% of energy per bit with respect to S_{ds}^{ZSS}-CARP when the packet size is 20B and up to 3% and to 8% when the packet size is 600B and 200B, respectively. These results are presented in Figure 3b and Figure 3c.

End-to-end latency. Figure 4 reports the average end-to-end latency of the packets correctly delivered to the sink by the four protocols for the three different packets sizes. As expected, when the packet size is 20B (Figure 4a), CARP experiences the lowest latency since the packet and digital signature sizes are comparable. S_{ds}^{ZSS}-CARP results in the highest latency since the overhead introduced is greater than that of S_{ds}^{Q}-CARP and it is comparable to that of S_{ds}^{BLS}-CARP but without the capability to aggregate the signatures. The latency experienced by S_{ds}^{ZSS}-CARP, S_{ds}^{BLS}-CARP and S_{ds}^{Q}-CARP increases by about 7%, 6% and 4% with respect to CARP for all the considered traffic loads. Figure 4b and Figure 4c show that when the packet size increases, all the protocols deliver the packets with higher latency due to the larger transmission times. In such a case, the overhead introduced by each digital signature scheme becomes negligible with respect to the time needed for actual data packet transmission. Therefore when considering packet size of 600B (200B), the latency experienced by S_{ds}^{ZSS}-CARP, S_{ds}^{Q}-CARP and S_{ds}^{BLS}-CARP increases by about 4%, (4.7%), 1%, (3.3%) and 0.5%, (3.6%), respectively, with respect to CARP for all the considered traffic loads, which leads to very limited latency degradation.

VII. Conclusions

In this paper we introduced SecFUN, a security framework for underwater acoustic sensor networks. SecFUN implements as building blocks AES in Galois Counter Mode (GCM) and short digital signature algorithms such as BLS, ZSS and Quartz to provide data confidentiality, integrity, authentication and non-repudiation. As a proof of concept, we extended the implementation of the Channel-aware Routing Protocol (CARP) to evaluate the proposed cryptographic primitives and their impact on the performance of the protocol, such as energy



(a) 20 Bytes (b) 200 Bytes (c) 600 Bytes

Figure 4: S_{ds}-CARP - End-to-end latency with different digital signatures schemes.

consumption and end-to-end latency. Results confirm that a flexible and full-fledged security solution tailored to meet the requirements of UASNs can be provided at a reasonable cost in terms of energy consumption and latency.

ACKNOWLEDGMENTS

This work has been partially supported by the EU FP 7 ICT project SUNRISE "Sensing, monitoring and actuating on the UNderwater world through a federated Research InfraStructure Extending the Future Internet" and by the PRIN project TENACE. In addition, Ateniese is supported in part by EU HORIZON 2020 projects Gains and SunFish.

REFERENCES

[1] J. Heidemann, M. Stojanovic, and M. Zorzi, "Underwater sensor networks: Applications, advances and challenges," *Philosophical Transactions of the Royal Society A*, vol. 370, pp. 158–175, August 2 2012.

[2] T. Melodia, H. Khulandjian, L.-C. Kuo, and E. Demirors, "Advances in underwater acoustic networking," in *Mobile Ad Hoc Networking: Cutting Edge Directions*, S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, Eds. Hoboken, NJ: John Wiley & Sons, Inc., March 5 2013, ch. 23, pp. 804–852.

[3] M. Domingo, "Securing underwater wireless communication networks," *IEEE Wireless Communications*, vol. 18, no. 1, pp. 22–28, February 2011.

[4] E. Souza, H. Wong, I. Cunha, A. Loureiro, L. Vieira, and L. Oliveira, "End-to-end authentication in under-water sensor networks," in *Proceedings of the* 18*th IEEE International Symposium on Computers and Communications (ISCC 2013)*, Split, Croazia, July 2013, pp. 000 299–000 304.

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

[6] S. Basagni, C. Petrioli, R. Petroccia, and D. Spaccini, "CARP: A channel-aware routing protocol for underwater acoustic wireless networks," *Ad Hoc Networks, Special Issue on Advances in Underwater Communications and Networks*, Available on-line, August 2014.

[7] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," *Journal of Cryptology*, vol. 17, no. 4, pp. 297–319, 2004. [Online]. Available: http://dx.doi.org/10.1007/s00145-004-0314-9

[8] F. Zhang, R. Safavi-Naini, and W. Susilo, "An efficient signature scheme from bilinear pairings and its applications," in *Public Key Cryptography–PKC 2004*. Springer, 2004, pp. 277–290.

[9] N. T. Courtois, M. Daum, and P. Felke, "On the security of hfe, hfev- and quartz," in *Public Key CryptographyPKC 2003*. Springer, 2002, pp. 337–350.

[10] C. Petrioli, R. Petroccia, J. R. Potter, and D. Spaccini, "The SUNSET framework for simulation, emulation and at-sea testing of underwater wireless sensor networks," *Ad Hoc Networks, Special Issue on Advances in Underwater Communications and Networks*, Available on-line, August 2014.

[11] M. B. Porter, "The BELLHOP manual and user's guide: Preliminary draft," La Jolla, CA, 2011, heat, Light, and Sound Research, Inc.

[12] M. C. Domingo, "Securing underwater wireless communication networks," *Wireless Communications, IEEE*, vol. 18, no. 1, pp. 22–28, 2011.

[13] Y. Cong, G. Yang, Z. Wei, and W. Zhou, "Security in underwater sensor network," in *Communications and Mobile Computing (CMC), 2010 International Conference on*, vol. 1. IEEE, 2010, pp. 162–168.

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

[15] D. Galindo, R. Roman, and J. Lopez, "A killer application for pairings: authenticated key establishment in underwater wireless sensor networks," in *Cryptology and Network Security*. Springer, 2008, pp. 120–132.

[16] R. Sakai, K. Ohgishi, and M. Kasahara, "Cryptosystems based on pairing," in *Proceedings of the 17th Symposium of Cryptography and Information Security, SCIS*, 2000, pp. 26–28.

[17] R. Barbulescu, P. Gaudry, A. Joux, and E. Thom, "A quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic," Cryptology ePrint Archive, Report 2013/400, 2013, http://eprint.iacr.org/.

[18] G. Dini and A. Lo Duca, "Optimized self organized sensor networks," *Sensors*, vol. 12, no. 11, pp. 15 133–15 158, 2012.

[19] Y. Zhang and Y. Zhang, "Wormhole-resilient secure neighbor discovery in underwater acoustic networks," in *Proceedings of INFOCOM 2010, IEEE*, March 2010, pp. 1–9.

[20] W. Wang, J. Kong, B. Bhargava, and M. Gerla, "Visualisation of wormholes in underwater sensor networks: a distributed approach," *International Journal of Security and Networks*, vol. 3, no. 1, pp. 10–23, 2008.

[21] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 3rd ed. Pearson Education, 2002.

[22] G. Bianchi, A. T. Capossele, C. Petrioli, and D. Spenza, "AGREE: exploiting energy harvesting to support data-centric access control in WSNs," *Ad hoc networks*, vol. 11, no. 8, pp. 2625–2636, 2013.

[23] G. Bianchi, A. T. Capossele, A. Mei, and C. Petrioli, "Flexible Key Exchange Negotiation for Wireless Sensor Networks," in *Proceedings of the fifth ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*. Chicago, Illinois, USA: ACM, 2010.

[24] G. Ateniese, G. Bianchi, A. T. Capossele, and C. Petrioli, "Low-cost Standard Signatures in Wireless Sensor Networks: A Case for Reviving Pre-computation Techniques?" in *Proceedings of the 20th Annual Network & Distributed System Security Symposium, NDSS'13*, San Diego, CA, USA, 2013.

[25] A. T. Capossele, V. Cervo, G. De Cicco, and C. Petrioli, "Security as a CoAP resource: an optimized DTLS implementation for the IoT," in *Proceedings of ICC 2015, IEEE*, London, UK, June 2015.

[26] The VINT Project, *The ns Manual*. http://www.isi.edu/nsnam/ns/, 2002.
[27] F. Guerra, P. Casari, and M. Zorzi, "World ocean simulation system (WOSS): A simulation tool for underwater networks with realistic propagation modeling," in *Proceedings of ACM WUWNet 2009*, Berkeley, CA, 3 November 2009, pp. 1–8.
[28] "World ocean atlas," www.nodc.noaa.gov/OC5/WOA05/pr_woa05.html.
[29] "General bathymetric chart of the oceans," www.gebco.net.
[30] "National geophysical data center, seafloor surficial sediment descriptions," http://www.ngdc.noaa.gov/mgg/geology/deck41.html.
[31] "TPS65950 Integrated Power Management IC." [Online]. Available: http://www.ti.com/product/tps65950