End-to-End Authentication in Under-Water Sensor Networks

Evaldo Souza, UFMG, Brazil, evaldoms@dcc.ufmg.br
Hao Chi Wong, Intel Corporation, hao-chi@intel.com
Ítalo Cunha, UFMG, Brazil, cunha@dcc.ufmg.br
A. A. F. Loureiro, UFMG, Brazil, loureiro@dcc.ufmg.br
L. F. M. Vieira, UFMG, Brazil, lfvieira@dcc.ufmg.br
Leonardo B. Oliveira, UFMG, Brazil, leob@dcc.ufmg.br

***Abstract*—Under-Water Wireless Sensor Networks (UWSNs) are a particular class of Wireless Sensor Networks (WSNs) in which sensors are located, as the name suggests, underwater. Applications of UWSNs range from oceanographic data collection to disaster prevention. UWSNs are vulnerable to attacks and because of their idiosyncrasies, security solutions for ground WSNs might not be applicable underwater. As a result, there is a need for mechanisms exclusively tailored to underwater environments. In this work we address the problem of authentication in UWSNs. We evaluate energy costs for different digital signature schemes for end-to-end authentication and discuss the tradeoffs involved in a number of scenarios. Our results show that schemes that perform well in ground WSN do not necessarily do well in UWSNs; and shed light on characteristics of a digital signature scheme that make them particularly suited to underwater networks.**

***Index Terms*—Aggregate Signatures, Authentication, Digital Signature Schemes, Security, Under-Water Sensor Networks.**

I. Introduction

Wireless Sensor Networks (WSNs) are ad-hoc networks comprised mainly of tiny sensor nodes with limited resources and one or more base stations (BSs), which connect the sensor nodes to the rest of the world [1]. Their applications range from battlefield reconnaissance and emergency rescue operations to surveillance and environmental protection.

Under-Water Wireless Sensor Networks (UWSNs) [2], in turn, are a particular class of WSNs in which sensors are deployed underwater. They allow a large range of applications, ranging from oceanographic data collection and offshore exploration to disaster prevention and assisted navigation. They differ from ground WSNs in a myriad of aspects. For instance, UWSNs are more prone to failures, their nodes' battery have less chances of being replaced, and they communicate through acoustic signals rather then through electromagnetic waves.

WSNs are vulnerable to attacks [3], and so are UWSNs [4]. UWSNs require security solutions since they may, for example, be used to monitor mineral and oil exploration and in turn carry sensitive information about a nation's natural resources. A number of security solutions have been proposed for ground WSNs [5], [6], but they are mostly not applicable to UWSNs [4]. As a result, there is a need for security mechanisms tailored exclusively to UWSNs. (In fact, UWSNs require not only tailored security mechanisms, but also other tailored components, such as protocol stack and hardware [2].)

Authentication plays a very important role in the context of security. An authentication mechanism may enables parties to discern between bogus and legitimate data, as well as allows parties to know with certainty who originated a given message. In WSNs and UWSNs, specifically, authentication also enables access control and mitigates Denial of Service (DoS) attacks [3].

As with confidentiality, authentication can be achieved through the use of symmetric or asymmetric cryptosystems. Asymmetric cryptosystems— or Public-Key Cryptosystems (PKCs)—however, usually provide more security properties and flexibility to users [7]. For instance, in the context of authentication, digital signatures provide an easy way for parties that have not previously interacted with one another to authenticate each other. One example is when UWSN nodes need to send authenticated data to transient ships passing through. Signatures are specially useful when entities switch communicating partners often. Digital signatures are also the only primitive that provides nonrepudiation, i.e., it prevents one node from denying previous commitments or actions [7].

Contribution. In this work we evaluate the power efficiency of three different digital signature schemes in UWSNs. We compare the Elliptic Curve Digital Signature Algorithm (ECDSA), the Zhang-Safavi-Naini-Susilo [8] (ZSS), and the Boneh-Lynn-Shacham [9] (BLS) schemes in UWSNs. Our main contributions are:

1) we evaluate some of the most popular digital signature schemes in terms of power consumption for various UWSN scenarios, highlighting their tradeoffs and pointing out the most adequate for each scenario; aggregate signatures, in particular, have not been evaluated for any WSN (including ground networks) before;
2) we contrast a ground WSN scenario against an UWSN scenario, and show how a given signature scheme performs differently in each case;
3) we draw and discuss conclusions on what makes a digital signature scheme attractive for UWSNs (in terms of power efficiency).

Note that digital signature schemes have already been evaluated in ground WSNs [6]. However, as we shall see (Section IV), their conclusions are not applicable to UWSNs.

Organization. The remainder of this work is organized as follows. In Section II, we discuss related work. In Section III, we discuss our evaluation methodology. In Section IV, we present results. Finally, in Section V we draw conclusions.

II. Brief Overview and Related Work

Ground WSNs and UWSNs differ in a myriad of aspects. For instance, compared to ground networks, in the UWSNs context: (i) communications are based on acoustic links (since the high energy absorption of water makes radio waves ill-suited to underwater communication); (ii) communication is significantly more expensive (the power needed for acoustic underwater communications is higher); (iii) nodes' battery have considerably less chances of being replaced (nodes access/ retrieval from under the water is labor intensive); (iv) are more prone to failures (because of fouling, corrosion, danger of getting wet and damaging components, etc.); (v) latency is higher (due to channel characteristics); (v) and bandwidth is severely limited (for due to a combination of many channel aspects)[2].

Like any WSNs, UWSNs are vulnerable to attacks [4], [10]. But because of the aforementioned UWSNs' idiosyncrasies, aside from a few exceptions (for instance, Galindo *et al.* [11] have solved the *key agreement problem* [7] for UWSNs by using the same strategy used for ground WSNs [12]), security solutions for ground WSNs (e.g., [5], [13], [14], [6]) are not applicable to or cannot be used as-is in UWSNs. For instance, proposals to thwart wormhole attack [1] in ground WSNs cannot be directly applied to UWSNs [10], [15], [16]. On the other hand, techniques that are immaterial in WSNs can be crucial in UWSNs. For example (Section IV) while the "short signature effect" is not so beneficial in ground WSNs, it can make a significant difference in UWSNs. This is so because acoustic underwater communication is extremely expensive, and differences in signature length can greatly impact the network's power consumption. Thus, research on novel mechanisms tailored exclusively to UWSNs is needed.

The size of research literature on security for UWSNs is quite small compared to that for WSNs. addressing the challenge of protecting them so far Kong *et al.* [10] discussed UWSNs' vulnerability to security attacks, as compared to ad-hoc networks and ground sensor networks. They specially focused on wormhole attacks, showing that UWSNs can be disrupted by wormholes of any length (in contrast to one-hop wormholes typically assumed in ground WSNs). They also proposed a localization service to detect and isolate underwater wormholes; however, the scheme is not practical because it incurs significant overhead. Domingo [4] identified characteristics of UWSNs that make them particularly vulnerable, and compiled a set of challenges to coming up with security solutions. They outlined attacks, listed security requirements, and suggested research questions. They kept the discussion at a high-level, and do not delve into details of possible solutions. Wang *et al.* [15] proposed Dis-VoW, a distributed mechanism to detect wormhole attacks using multidimensional analysis. Dis-Vow makes every node plot local network layout using multi-dimensional scaling and subsequently look for distortions in edge lengths and angles among neighbouring sensors which may indicate the existence of wormholes. Hu *et al.* [17] proposed WATERSync, a secure clock synchronization mechanism. To make WATERSync secure against insider attacks, network nodes observe time-stamp data originated from their neighborhood and apply a correlation test over them. Outlier data in test results may indicate the existence of wormholes. Zhang *et al.* [16]'s proposal also target wormhole attacks. They have used a different approach and do not rely on accurate clock synchronization to detect attacks. Instead, they leverage the fact that it is feasible to estimate the real acoustic signals' direction of arrival (ASDAs). To detect attacks, nodes contrast ASDAs estimates against real incoming signals. Galindo *et al.* [11] show how to solve the key agreement problem in UWSNs by using an authenticated non-interactive identity-based protocol. This strategy was used to solve the problem in ground WSN[12], but because of its non-interactive nature (i.e., communication is not required), it performed even better in the underwater context. Finally, to our knowledge, there is no work on digital signature schemes for UWSNs. There have been studies (e.g., [6], [18]) on signature schemes for ground WSNs, but they do not consider underwater characteristics, and then their evaluations are not applicable to UWSNs.

[1] A type of replay attack that tunnels messages from one point in the network to another over a low-latency link [3].

III. Methodology

This section describes the methodology used to evaluate signature schemes for end-to-end authentication in UWSNs. We discuss the signature schemes (Section III-A) as well as their costs in terms of energy usage (Section III-B).

A. Digital Signature Schemes

Computation required for signature generation is inversely proportional to the hardness of their underlying security problems [7]. As a result, traditional signature schemes (e.g. DSA/RSA [19]) based on subexponential problems demand a great deal of computation and are not adequate for resource-constrained devices such as sensor nodes [5]. Instead, Elliptic Curve Cryptography (ECC) schemes (e.g. ECDSA [19]) are used. Their underlying problem is fully-exponential and their signatures can be generated much more quickly [19].

There are different classes of ECC signature schemes. For instance, there are (i) those able to leverage on special parameters to speed computation up (e.g. Elliptic Curve Digital Signature Algorithm [19] – ECDSA) [19]; and (ii) those that produce shorter signatures (e.g. the Zhang-Safavi-Naini-Susilo [8] – ZSS). Among "short signature" schemes, there is also the Boneh-Lynn-Shacham [9] (BLS), which also belongs to the class of aggregate signatures schemes. These schemes have the additional capability of aggregating – combining – different signatures from different signatories (or not) into a single one. The result is referred to as aggregate signature. In BLS, aggregation is very efficient and the resulting signature has the same length as the original ones (aggregate and non-aggregate signatures have the same bit-length).

In this work we evaluate a representative of each of these classes, respectively ECDSA, ZSS, and BLS. We have chosen these schemes because: (i) ECDSA has become the de facto

Scheme	Generation	Signature size
ECDSA	134ms	40 bytes
ZSS	229ms	21 bytes
BLS	302ms	21 bytes

TABLE I: Schemes's generation times and signature lengths

standard in ECC; (ii) ZSS has been shown to be very computationally efficient in resource-constrained platforms; and (iii) to our knowledge, BLS is the only existing ECC aggregate signature scheme. Table I summarizes timings and signature sizes for the aforementioned schemes.

Our evaluation considers 80-bit security level – equivalent in strength to RSA-1024. We only address the problem of end-to-end authentication in this work. Because of signature length, signature schemes are not ideal for link-layer (hop-by-hop) authentication in networks with resource-constrained elements. Instead, Message Authentication Codes (MACs)[2] can be used [5]. MAC schemes are computationally efficient; and for a given security level, the various MAC schemes require approximately the same amount of processing, and produce equivalent size MACs. Thus, their analysis would not impact our conclusions about using signatures underwater.

Finally, note that we only evaluate the cost of signature generations. Because of the asymmetric nature of communication in WSNs, i.e. almost always from nodes to the BS [20], signature verification would be carried out by a resource-rich node for which differences in energy consumption level would be irrelevant.

B. Authentication Cost

In what follows, we describe how we quantify the cost of using signature schemes for end-to-end authentication in UWSNs. As usual in WSN security literature (e.g. [5]), we measure cost in terms of energy, the most constrained resource in WSNs [1]. We first discuss our network model (Section III-B1), then present how signing (Section III-B2) and transmission/reception (Section III-B3) costs were derived.

1) Network Model: we consider networks with frames that have header size $S_{\rm hdr}$ and a maximum payload size of $S_{\rm pay}$. Each node in the network sends messages to the BS. Messages are $S_{\rm msg}$ bytes long and are composed of $S_{\rm data}$ bytes of sensor data plus an optional signature of $S_{\rm sig}$ bytes. To send a (application-level) message of X bytes, the network needs to send $B(X)$ bytes, which depends on the number of full frames that need to be sent, $\lfloor X/S_{\rm pay} \rfloor$, and the amount of data on the last frame:

$$B(X) = \left\lfloor \frac{X}{S_{\rm pay}} \right\rfloor (S_{\rm hdr} + S_{\rm pay}) + (S_{\rm hdr} + X \bmod S_{\rm pay}).$$

We ignore interference, retransmissions, and ACKs on the link-layer. In practice, interference and retransmissions would incur extra communication overhead in the formulas below.

[2]We follow a standard practice in network security literature [5] and use *MAC* to denote Message Authentication Codes. We use link-layer to refer to Medium Access Control layer.

VAR	DEFAULT	Description
$S_{\rm data}$	210 B	Data size
$S_{\rm sig}$	varies	Signature size
$S_{\rm msg}$	varies	Message size ($S_{\rm data} + S_{\rm sig}$)
$S_{\rm pay}$	250 B	Maximum payload size
$S_{\rm hdr}$	10 B	Header size
$C_{\rm tx}$	562μJ/bit	(Re)Transmission cost
$C_{\rm rx}$	188μJ/bit	Reception cost
$C_{\rm sign}$	varies	Signing cost
h	6	Routing tree height
f	2	Routing tree fan-out
$B(X)$	—	Total number of bytes to send a message of X bytes

TABLE II: Notation

However, interference and retransmissions are independent of the cryptographic method used to sign messages in a UWSN. Their induced overhead does not depend on and is the same for all cryptographic solutions. Thus, this simplification does not impact the generality of our analysis.

We consider multi-hop UWSNs using a routing protocol that forms a forwarding tree rooted at the BS. We consider that the tree has height h, numbered from zero to $h-1$, and that each non-leaf node has k children. Messages from nodes in level i have to traverse i levels before reaching the BS. Thus, the total energy cost to transmit one message from a node n in level i to the BS is

$$c(n,i) = B(S_{\rm data} + S_{\rm sig})[iC_{\rm tx} + (i-1)C_{\rm rx}] + C_{\rm sign}. \quad (1)$$

We assume that time is slotted in one-hour *epochs*, i.e., the time between two consecutive samplings, and that each node sends one message to the BS in each epoch. The total transmission cost in one epoch is then

$$C_{\rm simple} = \sum_{i=1}^{h-1} \sum_{n=1}^{k^i} c(n,i),$$

where i iterates over all levels of the forwarding tree and n iterates over all nodes in a level.

The equations above can be used to compute the costs for a network without end-to-end authentication by simply setting $S_{\rm sig}$ and $C_{\rm sign}$ to zero. We focus on the scenario where all nodes send the same amount of data to the BS, i.e., $S_{\rm data}$ is the same for all nodes. However, this can easily be generalized by making $S_{\rm data}$ variable in Eq. (1).

With signature aggregation (i.e. BLS), a node can forward any number of messages and send a single aggregate signature. Note that, with signature aggregation, a signature is transmitted once then aggregated with other signatures. The cost to send a message from a node n in level i to the BS is

$$\begin{aligned} c_{\rm agg}(n,i) =& B(S_{\rm data})[iC_{\rm tx} + (i-1)C_{\rm rx}] + C_{\rm sign} + \\ & B'(S_{\rm sig})[C_{\rm tx} + C_{\rm rx}], \end{aligned} \quad (2)$$

where the first line is Eq. (1) excluding the cost to transmit the signature, and the second line is the cost to transmit the signature to the next hop. Function B' computes the number of bytes required to transmit the signature, which may include an extra header if $S_{\rm data} \leq S_{\rm pay}$ and $S_{\rm data} + S_{\rm sig} > S_{\rm pay}$. Note

signature aggregation is very computational efficient [3] [19] and for simplicity we ignore its processing costs in our evaluation.

The total cost to send messages to the BS in one epoch using signature aggregation is then

$$C_{\text{agg}} = \sum_{i=1}^{h-1} \sum_{n=1}^{k^i} c_{\text{agg}}(n, i).$$

2) Signing Costs: Signing costs C_{sign} and C_{agg} in terms of energy can be derived from the duration of computation and hardware characteristics. To estimate energy consumption, we take into account the time for signature generation as well as current and voltage values for the target platform. Time was obtained by running publicly available[4] signature scheme implementations on the MSP430F2418 (16-bit/16MHz and 116KB *flash*[21]) platform. We chose this processing unit because it is energy-efficient, and has been previously used for underwater node designs (e.g., in [22]).

3) Transmission & Reception Costs: Transmission and reception costs, C_{tx} and C_{rx}, depend on the hardware as well as the physical- and the link-layer.

Concerning the link-layer, current UWSN research focuses on CSMA and CDMA solutions [23]. (FDMA is not suitable because of acoustic channel's narrow bandwidth, and TDMA is not suitable because it requires long time guards due to the long propagation delay in underwater acoustic channel [23].) We use values from an established CDMA protocol for underwater environments [24]. The frame payload size (S_{pay}) and header size (S_{hdr}) are 250 and 10 bytes respectively.

We adopt the energy consumption model given in [25]:

$$E = P_0 T_p d^k a^d \tag{3}$$

where E denotes the transmission energy, P_0 denotes the reference reception power necessary to decode the received frame, T_p is the frame transmission time, d is the distance from the source to the destination, k is the spreading factor (which is 2 for spherical spreading), $a = 10^{\alpha(f)/10}$ is a frequency-dependent term obtained from the absorption coefficient $\alpha(f)$, which is given by Thorp's expression [25] as:

$$\alpha(f) = 0.11\frac{f^2}{1+f^2} + \frac{44f^2}{4100+f^2} + 2.75\times10^{-4}f^2 + 0.003 \tag{4}$$

where f is in kilohertz. (For more on Thorp's expression, see [25].)

We instantiated this model using the WHOI modem[5] with a throughput of 480bit/s and a range of 25m. As a result, C_{tx} and C_{rx} are 562μJ/bit and 188μJ/bit respectively.

IV. Performance Evaluation

This section presents results on the performance of signature schemes underwater. We evaluate them by contrasting transmission and signing costs (Section IV-A), short and aggregate signature effects (Section IV-B), delay sensitiveness of applications (Section IV-C), and finally ground WSNs and UWSNs (Section IV-D). Unless otherwise noted, we use the default parameter values shown in Tab. II.

[3]Signature aggregation is equivalent to an elliptic curve point addition.
[4]We use the RELIC toolkit: http://code.google.com/p/relic-toolkit/
[5]http://acomms.whoi.edu/umodem

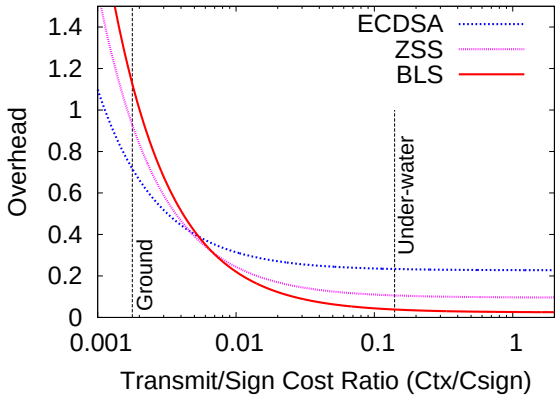


Fig. 1: Transmission versus signing costs.

A. Transmission versus signing costs

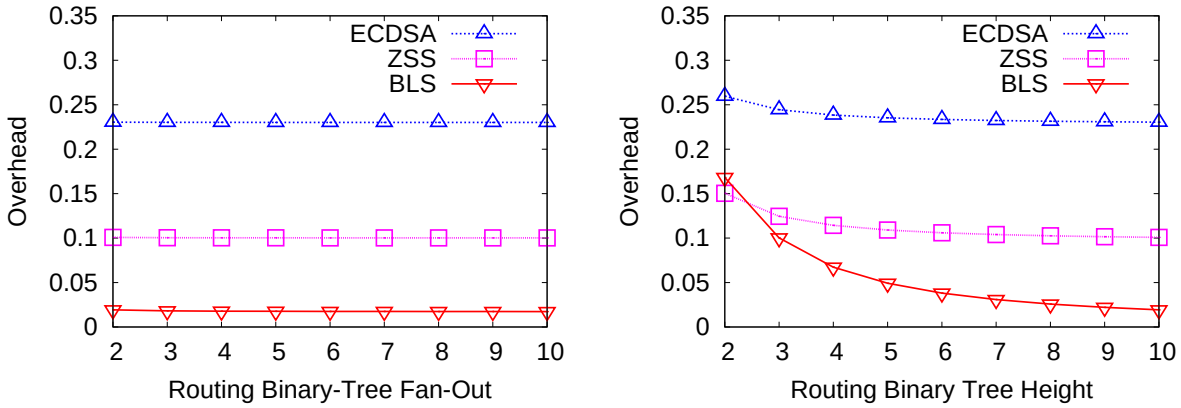
One key factor when choosing an authentication scheme is the ratio between transmission (C_{tx}) and signing (C_{sign}) costs. If the modem requires more energy than the CPU, then it is probably better to spend extra CPU cycles to compute a short and/or aggregate signature. Conversely, if the CPU requires more energy than the modem, it is probably better to to transmit longer signatures and save CPU cycles. Our models presented in Sec. III allow us to compute what signing scheme is most appropriated to a UWSN based on its transmit/sign cost ratio.

Fig. 1 shows the whole network overhead induced by end-to-end authentication when compared to plain transmissions, i.e. without signatures. We vary the ratio between transmission and signing costs on the x-axis. We set the cost to generate an ECDSA signature (the cheapest to compute) to 1 and normalize other schemes' signing costs based on ECDSA's.

In scenarios where the transmission cost is significantly lower than the signing cost (shown on the left side of Fig. 1), ECDSA presents the lowest overhead. This is because power savings from ECDSA's faster signature generation outweigh savings from BLS's and ZSS's shorter signature sizes. BLS incurs the highest overhead because it is the most computationally intensive, and computation is much more expensive than communication (Table I). This is the typical scenario in ground WSNs (see the "grounded" vertical line in Fig. 1).

On the right side of the graphs, where transmission cost is higher, short and aggregate signature schemes incur lower overhead. BLS has the lowest overhead since it has both short and aggregate signatures.

Interestingly, we see that there is a small region (for ratios around 0.004 and 0.006) where ZSS beats both ECDSA and BLS (by a short margin). This is the region where BLS's additional signing cost compared to ZSS's is higher than savings due to its aggregation capability; and where ECDSA's



(a) Overhead vs. Fan-out

(b) Overhead vs. Height

Fig. 2: Network topology and authentication overhead.

additional transmission cost compared to ZSS's is higher than savings induced by its quicker signature generation.

Note that our default parameters (Table II) produce a transmit/sign cost ratio of 0.14. For this particular ratio, BLS is the most efficient option.

B. Short and aggregate signatures versus overhead

The impact of shorter and aggregate signatures on the overall overhead incurred by the authentication scheme can be observed when the network topology varies. For instance, Fig. 2a presents end-to-end authentication overhead as a function of the routing binary tree fan-out. It shows that, for a given tree-height, the fan-out has no impact on overhead regardless of the scheme.

No matter the fan-out, the overhead incurred by both computation and communication is fixed. That is, for each additional node there is one more data block being signed and a message being sent via a route with the same number of hops. This in turn incurs the same number of (re)transmissions per node and therefore the overhead maintains constant.

Fig. 2b shows that changing the height of the UWSN routing binary-tree has an impact on ZSS and ECDSA. This is because the overhead incurred by transmitting the signatures increases with the tree height, while signing costs remain the same (i.e., one signing per node). The small decrease in ZSS and ECDSA is due to the amortization of signing costs as the overall transmission/reception costs increase. As signing costs become small compared to signature transmission costs, ZSS's and ECDSA's overhead converge to the overhead of transmitting the signatures. BLS's signing cost suffers from the same amortization process. However, BLS's overhead keeps decreasing since the number of signature transmissions/receptions does not increase with route length due to aggregate signatures.

C. Delay-sensitive versus delay-insensitive applications

Delay-sensitive or real-time applications need data to be sent as soon as they are available. Applications that are delay-insensitive, however, have no time restrictions and can afford to amass multiple samplings before transmitting them to the BS. To show the impact of application behavior and requirements on authentication overhead, Fig. 3 shows overhead as a function of data size (S_{data}). Delay-sensitive applications may have to send data in small chunks (left side of the figure), while delay-insensitive applications may amass data to send large chunks (right side).

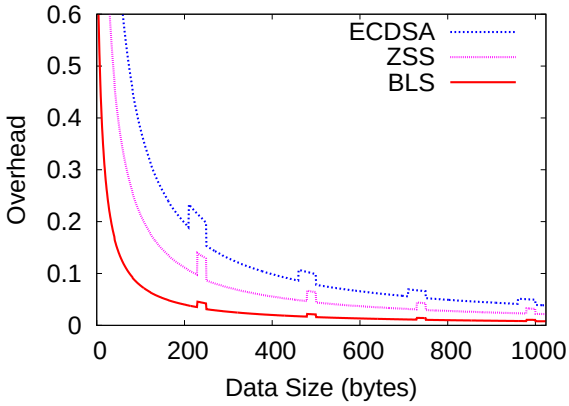


Fig. 3: Data size and authentication overhead.

Unlike in link-layer authentication solutions, where overhead does not depend on data size [5], the overhead incurred by end-to-end authentication mechanisms decreases as data size increases. In the former, messages are broken into frames and each frame carries a MAC that induces overhead. In the latter, a single signature of fixed size is appended to the data block regardless of data size. [6] Signing costs are virtually [7] unaltered as well. This is because the signing operation acts in fact over the message's hash value, which is of a fixed size. This makes signature schemes "cheaper" for delay-insensitive

[6]We are not arguing that application-layer authentication schemes are more adequate than link-layer schemes. We believe the solutions are complementary.

[7]The cost of a cryptographic hash computation is proportional to message size, but the operation is efficient and its costs negligible in this context [7].

applications.

Note that there are spikes in Fig. 3. A spike begins right after the authenticated transmission starts to require an additional frame transmission to send a message, but the plain transmission does not. (I.e., whenever $S_{\mathrm{data}} + S_{\mathrm{sig}}$ becomes larger than S_{pay}. This incurs more overhead since an extra frame header must be sent.) And the spike ends right after the plain transmission also starts to require an additional frame transmission to send a message (i.e., whenever S_{data} alone becomes larger than S_{pay}).

D. Ground versus underwater networks

Digital signature schemes in ground WSNs have been studied before (e.g., [6]). Even though these studies assumed different models and protocol stacks (as compared to what we assumed in this paper), we can still draw some comparisons.

Taking into account an entire network, transmit power are often significantly higher than (active) CPU power in WSNs – approximately $12\times$ for the Telos node [21]. Thus, one would expect "short signature" schemes to be more energy-efficient in WSNs. In [6], however, the authors showed that this is not always the case. In their ground WSN over TCP/IP, for instance, ECDSA beats BLS despite the former's longer signatures. This is because, in their scenario, the computation is orders of magnitude longer than transmission, and the overhead is mainly because of signing costs. This makes ECDSA, computationally the fastest scheme, the most energy-efficient in that context. It is true that they considered a single-hop scenario, which favors ECDSA to the detriment of other schemes. However, ECDSA is so much more efficient than others (in fact, approximately $5\times$ [6]) that it would beat the others even in scenarios with a few hops.

This is not the case in UWSNs. In these networks, both short and aggregate signature schemes have better performance. The ratio of communication and computation costs in UWSNs is much higher than in WSNs. In UWSNs, reducing communication costs is paramount.

V. Conclusion

UWSNs are a class of WSNs where sensors are deployed underwater. UWSNs are vulnerable to attacks and because of UWSNs' peculiarities, security solutions conceived for ground WSNs are not applicable. So, it is necessary to devise security solutions exclusively tailored to UWSNs. In this work, we evaluated power consumptions of different digital signature schemes underwater, and identified characteristics of a signature scheme that would make it well-suited to UWSNs. We discussed the tradeoffs involved in a number of scenarios, and showed how the use of both short and aggregate signatures can increase energy-efficiency in UWSNs.

References

[1] D. Estrin, R. Govindan, J. S. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *MobiCom'99*, pp. 263–270, 1999.

[2] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad Hoc Networks*, vol. 2, no. 3, pp. 257–279, 2005.

[3] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," in *First IEEE International Workshop on Sensor Network Protocols and Applications*, 2003.

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

[5] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: Security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002. Also in MobiCom'01.

[6] L. B. Oliveira, A. Kansal, B. Priyantha, M. Goraczko, and F. Zhao, "Secure-TWS: Authenticating node to multi-user communication in shared sensor networks," in *International Conference on Information Processing in Sensor Networks IPSN'09*, pp. 289–300, 2009.

[7] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 2001.

[8] F. Zhang, R. Safavi-naini, and W. Susilo, "An efficient signature scheme from bilinear pairings and its applications," in *PKC 2004, LNCS 2947*, pp. 277–290, Springer-Verlag, 2004.

[9] D. Boneh, B. Lynn, and H. Schacham, "Short signatures from the Weil pairing," *Journal of Cryptology*, vol. 17, no. 4, pp. 297–319, 2004.

[10] J. Kong, Z. Ji, W. Wang, M. Gerla, R. Bagrodia, and B. Bhargava, "Low-cost attacks against packet delivery, localization and time synchronization services in under-water sensor networks," in *4th ACM workshop on Wireless security (WiSe'05)*, pp. 87–96, 2005.

[11] D. Galindo, R. Roman, and J. Lopez, "A killer application for pairings: Authenticated key establishment in underwater wireless sensor networks," in *7th International Conference on Cryptology and Network Security (CANS '08)*, pp. 120–132, 2008.

[12] L. B. Oliveira, M. Scott, J. López, and R. Dahab, "Tinypbc: Pairings for authenticated identity-based non-interactive key distribution in sensor networks," in *5th International Conference on Networked Sensing Systems (INSS'08)*, (Kanazawa/Japan), pp. 173–179, 2008.

[13] S. Zhu, S. Setia, and S. Jajodia, "LEAP: efficient security mechanisms for large-scale distributed sensor networks," in *10th ACM conference on Computer and communication security (CCS'03)*, pp. 62–72, ACM Press, 2003.

[14] L. B. Oliveira, A. Ferreira, M. A. Vilaça, H. C. Wong, M. Bern, R. Dahab, and A. A. F. Loureiro, "SecLEACH– on the security of clustered sensor networks," *Signal Process.*, vol. 87, no. 12, pp. 2882–2895, 2007.

[15] W. Wang, J. Kong, B. Bhargava, and M. Gerla, "Visualisation of wormholes in underwater sensor networks: a distributed approach," *Int. J. Secur. Netw.*, vol. 3, pp. 10–23, December 2008.

[16] R. Zhang and Y. Zhang, "Wormhole-resilient secure neighbor discovery in underwater acoustic networks," in *29th conference on Information communications (INFOCOM'10)*, pp. 2633–2641, 2010.

[17] F. Hu, S. Wilson, and Y. Xiao, "Correlation-based security in time synchronization of sensor networks.," in *IEEE Wireless Communications and Networking Conference (WCNC'08)*, pp. 2525–2530, 2008.

[18] C. P. L. Gouvêa, L. B. Oliveira, and J. López, "Efficient software implementation of public-key cryptography on sensor networks using the msp430x microcontroller," *Journal of Cryptographic Engineering*, vol. 2, pp. 19–29, 2012.

[19] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to elliptic curve cryptography*. Springer, 2004.

[20] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, pp. 102–114, August 2002.

[21] V. Handziski, J. Polastre, J.-H. Hauer, and C. Sharp, "Flexible hardware abstraction of the ti msp430 microcontroller in tinyos," in *SenSys '04: 2nd international conference on Embedded networked sensor systems*, (New York, NY, USA), pp. 277–278, ACM Press, 2004.

[22] D. Pinto, S. S. Viana, J. A. M. Nacif, L. F. M. Vieira, M. A. M. Vieira, A. B. Vieira, and A. O. Fernandes, "Hydronode: a low cost, energy efficient, multi purpose node for underwater sensor networks," in *37th IEEE Conference on Local Computer Networks (LCN)*, 2012.

[23] D. Pompili and I. F. Akyildiz, "Overview of networking protocols for underwater wireless communications," *IEEE Communications Magazine*, pp. 97–102, Jan. 2009.

[24] D. Pompili, T. Melodia, and I. F. Akyildiz, "A cdma-based medium access control protocol for underwater acoustic sensor networks," *IEEE Trans. on Wireless Communications*, vol. 8, no. 4, pp. 1899–1909, 2009.

[25] E. M. Sozer, M. Stojanovic, and J. G. Proakis, "Underwater Acoustic Networks," *IEEE Journal of Oceanic Engineering*, vol. 25, no. 1, pp. 72–83, 2000.