Physical Layer Authentication in Underwater Acoustic Networks with Mobile Devices

Paolo Casari
DISI, University of Trento, and CNIT
Trento, Italy
paolo.casari@unitn.it

Francesco Ardizzon
DEI, University of Padova, and CNIT
Padova, Italy
francesco.ardizzon@phd.unipd.it

Stefano Tomasin
DEI, University of Padova, and CNIT
Padova, Italy
stefano.tomasin@unipd.it

ABSTRACT

As underwater acoustic communication technologies become mature and underwater networks evolve into reliable solutions for data communications, authenticating transmitted data turns from an option to a necessity. In this paper, we explore physical-layer authentication for an underwater acoustic networks with mobile devices. We choose the power-weighted average of the channel taps' arrival delay as the main authentication feature. We then develop a Kalman filter approach to track the evolution of this feature in the presence of mobility. The filter computes an innovation metric for each new transmission, which is processed to determine if a signal originates from a legitimate network node. Simulation results obtained from a dataset generated with Bellhop show that our authentication mechanism successfully distinguishes between legitimate and impersonating transmitters. Moreover, we show that linearly combining innovation readings from multiple sensors yields a good low-complexity classifier, and assess the impact of the transmitter speed on the authentication performance.

CCS CONCEPTS

• **Security and privacy** → **Authentication**; • **Networks** → *Mobile networks*; • **Computing methodologies** → *Machine learning*.

KEYWORDS

Underwater acoustic networks, mobility, authentication, physical layer security, simulation

ACM Reference Format:
Paolo Casari, Francesco Ardizzon, and Stefano Tomasin. 2022. Physical Layer Authentication in Underwater Acoustic Networks with Mobile Devices. In *The 16th International Conference on Underwater Networks & Systems (WUWNet'22), November 14–16, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3567600.3567604

1 INTRODUCTION

Underwater acoustic networks (UWANs) employ acoustic waves to enable communications among devices under the sea surface. However, the harsh propagation environment typically limits the achievable data rates, and requires complex signal processing algorithms at the receiver to cope with long channel impulse responses, significant Doppler spread, as well as interference from other acoustic sources [33]. In such a scenario, the overhead imposed by security protocols operating at the higher layers of the protocol stack may become problematic. Thus, solutions working at the physical layer (going under the name of physical-layer security, PLS) have been a very active subject of research in recent years.

Consider the authentication problem, i.e., deciding if a received message comes from a legitimate transmitter or from an impersonating attacker. Conventional physical-layer security (PLS) authentication techniques typically use the channel impulse (or frequency) response estimated from the received signal as a *fingerprint* that is unique to the transmitter [18]. Thus, an initial received packet, whose authenticity has been established by higher-layer cryptographic techniques, provides the reference channel response. Subsequent packets (without a higher-layer authentication mechanism) are accepted as authentic when the newly estimated channel response matches the reference one.

However, in UWANs, devices are subject to continuous movements induced, among others, by drifting under the influence of sea waves and currents. Due to this and to the acoustic frequencies used, the variation of the channel over time can be significant. As a result, any assumption that the channel does not change across multiple transmissions becomes unrealistic.

In this paper, we propose a PLS authentication technique for underwater acoustic communications, specifically designed to account for the time variability of the channel. To this end, we consider the power-weighed average delay of the channel taps as the authentication parameter, because it is well related to the distance between the transmitter and the receiver. When such delay is measured from the same source to different cooperating receivers, it provides a robust signature of the transmitter. To take mobility into account, we apply a Kalman filter on the average delay, and track delay variations by assuming a simple linear evolution model with slowly varying velocity. The Kalman filter itself will estimate the instantaneous velocity to best track the delay variations. The authentication check is then obtained by comparing the *innovation* of the Kalman filter with a given threshold. Such innovation indicates the discrepancy between the Kalman-predicted value and the observation, and an irregular behavior of the observed delay indicates a possible attack.

The rest of the paper is organized as follows. Section 2 surveys related work on the authentication of underwater transmissions, including physical layer approaches; Section 3 describes the system model; Section 4 details our proposed authentication algorithm; Section 5 presents simulation results. Finally, Section 6 draws concluding remarks.

WUWNet'22, November 14-16, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9952-4/22/11…$15.00
https://doi.org/10.1145/3567600.3567604

2 RELATED WORK

Authentication is a key security functionality in UWANs as it is in terrestrial networks [17, 21]. Through authentication mechanisms, underwater network nodes can autonomously decide whether a received message has been sent by a legitimate network member, or rather by an attacker trying to impersonate a legitimate node.

Several methods have been investigated to achieve authentication in UWANs. The most straightforward approach is to consider classical cryptography algorithms that would work for terrestrial cabled or wireless networks and evaluate their impact on underwater applications. Souza *et al.* explore the communication and computation energy toll that terrestrial network authentication primitives may take if directly applied to underwater network nodes for end-to-end authentication [31]. The authors conclude that short and aggregate signature schemes are recommended in underwater networks.

The work in [11] takes a different approach, and instead proposes a secure protocol suite for UWANs. As a part of this suite, the authors advocate the use of message authentication codes [22] to preserve message integrity, even at the expense of increased message length. The survey in [24] proposes game theory as a means to foster cooperation among network nodes, by motivating them to improve the effectiveness of end-to-end authentication schemes, which are seen as a key functionality of future UWANs [29].

With the aim to reduce the complexity of underwater authentication, Yuan *et al.* employ matrices of known structure as part of the process, so as to reduce their memory occupancy and the computational cost of the authentication algorithm [36]. The proposed scheme achieves up to four orders of magnitude less complexity than the standard RSA-based authentication. With a similar purpose, Al Guqhaiman *et al.* propose a multi-factor scheme based on zero-knowledge proofs via message authentication codes [1]. Specifically, the codes depend not only on pre-shared information, but also on communication-related features such as the MAC address of the node, direction of arrival information, as well as the hop count of the sender. Receiving a packet for which this data does not match any of the features of the receiver's neighbors causes the receiver to label the packet as malicious, and to send an alert to its own network neighborhood.

Zhang *et al.*'s approach [38] revolves around classical authentication schemes based on message exchanges, and mandates the use of lightweight primitives such as chaotic maps and hash functions. While being slightly lighter than competing schemes from the literature from a computational point of view, the proposed scheme requires less storage to work. Along the same line, in [16] the attacker is able to impersonate multiple network nodes at once (also known as a Sybil attack). Here, the legitimate nodes attempt to identify the attacker's malicious behavior via its node id and the data stored in the cluster head, which feeds a hierarchical fuzzy system-based trust management model.

Recently, physical layer security approaches have been considered both for authentication and for other security primitives such as key exchange. Physical layer authentication often relies on the collection of channel characteristics (e.g., features of the channel impulse response) to tell apart transmissions by legitimate network members from transmissions by an impersonating attacker.

Considering an underwater LoS environment with negligible multipath, Khalid *et al.* propose that the receiver keep a database of angles of arrival for legitimate transmissions from a given node [20]. In this way, the receiver can detect an attacker by comparing the angle of arrival of its transmissions against the distribution of previously collected angles of arrival. The matching evaluation metric is the Mahalanobis distance. However, the work does not consider the case of a more powerful attacker that can craft transmitted signals to change the estimated angle of arrival at the receiver.

Aman *et al.* evaluate the capacity of underwater channels under impersonation attacks [2], assuming that the legitimate receiver uses distance as a feature to discriminate between a legitimate and an impersonating transmitter. After modeling the dynamics of the communications as a Markov chain, the authors numerically optimize the optimum transmission rate for the legitimate transmitter and show that a small neural network reproduces the optimization process well.

In [39], the authors propose to authenticate nodes based on a single feature, the maximum time-reversal resonating strength, which measures how well a received channel impulse response matches those of previous transmissions, stored in a pre-collected database. The authentication mechanism is then based on a Neyman-Pearson likelihood ratio test (LRT).

The work in [9] considers a large dataset of underwater channel feature measurements, and evaluates which features remain coherent over time while becoming uncorrelated already over short distances.[1] Assuming that several trusted nodes hear both the transmissions of a legitimate node and those of an attacker, the proposed method trains generalized Gaussian probability density functions (PDFs) to represent the features of legitimate transmissions. Then, it coordinates the trusted nodes to make a decision whether an incoming transmission obeys the previously learned statistics or not. The method is robust against attackers that can precode their transmission to change the channel perceived by multiple trusted nodes at the same time.

The above work was further extended in [4] to automatically extract the feature statistics using neural network, thereby avoiding the need to fit a generalized Gaussian PDF to the channel data in [3] to model the correlation among different features using autoencoder neural networks. Both local training and global training are considered. With local training, each trusted node makes a local decision on authenticity, and a sink uses a neural network to fuse the decisions, making it unnecessary to communicate anything other than the local decisions. Conversely, global training achieves better performance, but requires all trusted nodes to communicate the weights of their local neural networks.

In contrast with the existing literature, we do not directly exploit channel impulse response features to tell apart a legitimate node from an attacker in our physical layer authentication approach. Rather, we deploy a Kalman filter, which tracks the evolution of the power-weighed average of the delay of the most significant channel taps corresponding to a legitimate transmitter. We then use the innovation computed by the Kalman filter to discriminate

[1] A similar channel feature analysis and extraction, targeting secret key generation instead of authentication, was performed in [25, 26].

among a legitimate and an impersonating transmitter. This approach factors in mobility by design. Our results prove that the resulting scheme works even if the attacker can track and localize a legitimate transmitter with different degrees of accuracy, and can arbitrarily manipulate receiver-side impulse responses.

3 SYSTEM AND ATTACKER MODEL

We model Bob as a set of N closely deployed sensors, $\{S_1, \cdots, S_N\}$. By using his N sensors, Bob needs to decide whether a received packet comes from a legitimate transmitter (Alice), or from an attacker (Eve) attempting to impersonate Alice.

We assume that Alice is a mobile device, e.g., a drifter or an autonomous underwater vehicle (AUV), that periodically transmits information to Bob. Transmissions occur via underwater acoustic channels. Each sensor S_n is connected to Bob through a cable and any information transfer through the cables is error-free,[2] authenticated, and integrity-protected. In other words, Eve cannot interfere with the collection of data from the sensors to the logic making the authenticity decision (see Section 4). Because the decision is based on time information, in this paper we assume that Alice is synchronized with Bob (e.g., using one of the many schemes designed for underwater networks [6, 10, 35]) and leave the case where round-trip delays are estimated via (vulnerable) message exchanges as a future extension.

In these conditions, Eve can still attempt to impersonate Alice by crafting signals whose features appear similar to those of Alice's transmissions. To do so, we assume that Eve knows all details and parameters of the authentication algorithm, and that she is also synchronized with Alice and Bob. Moreover, Eve can precode her transmissions to reproduce any desired channel impulse response at any of Bob's sensors, including even crafting a different channel response for each sensor. We remark that the above capabilities imply perfect knowledge of the environment (e.g., the surface and bottom profile, as well as the sound speed profile in the network area), and require channel estimation, precoding, the availability of multiple transceivers, and considerable processing power (including computing multiple ray tracing outputs within a negligible amount of time). Thus, the above model is quite generous towards Eve. Finally, Eve does not know the exact location of Alice, but can localize her, e.g., using the well-known approaches of [13, 34], or matched field processing techniques [12, 23]. Eve's estimate of Alice's 3D location vector is then

$$\hat{\boldsymbol{P}}_{\text{Alice}} = \boldsymbol{P}_{\text{Alice}} + \boldsymbol{\epsilon} \,, \tag{1}$$

where $\boldsymbol{P}_{\text{Alice}}$ is the true location of Alice and $\boldsymbol{\epsilon}$ models the localization error.

4 PROPOSED AUTHENTICATION APPROACH

In our proposed algorithm, upon receiving a packet at time t:

(1) each sensor S_n estimates the power-delay profile $\{\Pi_n(t,\tau)\}$ (i.e., $\Pi_n(t,\tau)$ is the power of the tap with delay τ) and processes it to extract a feature $\hat{x}_n(t)$;
(2) each sensor S_n then exploits a previously trained model to predict Alice's feature $\tilde{x}_n(t)$; next, it compares the prediction $\tilde{x}_n(t)$ to the measured feature $\hat{x}_n(t)$, computing a model correction term $\beta_n \in \mathbb{R}$;
(3) Bob receives the corrections from all his sensors; to improve the performance of scheme, Bob can collect K observations per receiver, concatenated into vector

$$\boldsymbol{\beta} = [\beta_{1,1}, \cdots, \beta_{1,K}, \beta_{2,1}, \cdots, \beta_{2,K}, \cdots, \beta_{N,1} \cdots, \beta_{N,K}] \,. \tag{2}$$

(4) Finally, Bob computes the decision variable $\gamma = g(\boldsymbol{\beta})$, and tests the authenticity of the packet as

$$\hat{\mathcal{H}} = \begin{cases} 0, & \text{if } \gamma < \lambda \quad \text{(packet from Alice)}, \\ 1, & \text{if } \gamma \geq \lambda \quad \text{(packet from Eve)}. \end{cases} \tag{3}$$

Bob sets the threshold λ to achieve a target false alarm (FA) probability. Call $\mathcal{H} = 0$ ($\mathcal{H} = 1$) the case where Alice (Eve) is actually transmitting: the FA probability is $p_{\text{FA}} = \mathbb{P}[\hat{\mathcal{H}} = 1 | \mathcal{H} = 0]$, and the missed detection (MD) probability is $p_{\text{MD}} = \mathbb{P}[\hat{\mathcal{H}} = 0 | \mathcal{H} = 1]$. In the next sections, we describe how each step is implemented, discussing several strategies to design the function $g(\cdot)$.

4.1 Feature Extraction

We consider the *power-weighed average delay* as the authentication feature. To compute it, we first zero out low-power arrivals in the power-delay profile $\{\Pi_n(t,\tau)\}$, i.e.,

$$\Pi'_n(t,\tau) = \begin{cases} 0, & \text{if } \Pi_n(t,\tau) < T_h, \\ \Pi_n(t,\tau), & \text{if } \Pi_n(t,\tau) \geq T_h, \end{cases} \tag{4}$$

where T_h is chosen to obtain a desired FA probability when discriminating true arrivals from noisy contributions [8]. Call $\mathcal{H}_n(t)$ the set of delays of all channel arrivals that remain after thresholding. As pointed out in Section 3, we assume that all devices are synchronized, thus we can compute the average delay as

$$x_n(t) = \frac{1}{\bar{\Pi}_n(t)} \sum_{\tau \in \mathcal{H}_n(t)} \tau \Pi'_n(t,\tau) \,, \tag{5}$$

where

$$\bar{\Pi}_n(t) = \sum_{\tau \in \mathcal{H}_n(t)} \Pi'_n(t,\tau) \,, \tag{6}$$

and we remark that the delay of the first arrival in the channel impulse response depends on the distance between the transmitting and receiving devices. Note that it is still possible to extend the feature set without loss of generality using the candidate features discussed in [4, 30].

4.2 Prediction Strategy

We now describe the Kalman filter we employ to track the average delay feature and understand how much innovation a new transmission brings. To simplify the notation, we drop both the time reference and the sensor index n. Thus, we consider that each sensor collects a sequence of delay measurements $\{\hat{x}_i\}$, where measure $\hat{x}_i$ is associated with the power delay profile observed at time t_i. The task of the Kalman filter is to track the evolution of the distance d_i between S_n and Alice, and the projected velocity v_i, i.e., the velocity component along the direction of the LoS between S_n and Alice. Thus, the *(hidden) true state* at step i is $\boldsymbol{z}_i = [d_i, v_i]^{\text{T}}$.

[2] The very low error rate within the cables can be easily compensated for with standard mechanisms such as the use of cyclic redundancy checks (CRCs), and forward error correction (FEC).

Considering the previously described scenario, we relate subsequent observations of the distance and of the (projected) velocity using a *local linear movement model* with random evolution, i.e., defining the state transition matrix

$$\boldsymbol{F}_i = \begin{pmatrix} 1 & \Delta t_i \\ 0 & 1 \end{pmatrix} , \tag{7}$$

where $\Delta t_i = t_i - t_{i-1}$, the evolution of the true state is modeled as

$$\boldsymbol{z}_i = \boldsymbol{F}_i \boldsymbol{z}_{i-1} + \boldsymbol{w}_i , \tag{8}$$

where $\boldsymbol{w}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_i)$ represents the process noise, assumed to have Gaussian statistics.

About the observations, we first introduce the measurement vector as

$$\boldsymbol{h}_i = \begin{pmatrix} 1/v & 0 \end{pmatrix} , \tag{9}$$

where v is the sound speed in water. Then, the true state $\boldsymbol{z}_i$ is related to the measurement x_i by the following function

$$x_i = \boldsymbol{h}_i \boldsymbol{z}_i + r_i , \tag{10}$$

where $r_i \sim \mathcal{N}(\boldsymbol{0}, \sigma_i)$ models the observation noise. We remark that, in general, the receiver does not know the actual sound speed in water, also because taking a local sound speed measurement requires the deployment of possibly bulky equipment. Still, we can use an approximated value v, with the understanding that the term r_i in (10) also incorporates sound speed approximation errors.

We also remark that the Kalman filter assumes a Gaussian statistic for both $\boldsymbol{w}_i$ and r_i: while this hypothesis is not always true in our scenario, we still can consider it as an approximation. However, when the hypotheses are met and (8) and (10) perfectly model the reality, the Kalman filter is proven to be an optimal predictor [19].

For each received packet, the Kalman filter performs two operations: *prediction* and *model update*. During the prediction step, it computes the *a priori* state estimate $\tilde{\boldsymbol{z}}_{i|i-1}$ and its covariance matrix $\boldsymbol{P}_{i|i-1}$, respectively, as

$$\tilde{\boldsymbol{z}}_{i|i-1} = \boldsymbol{F}_i \tilde{\boldsymbol{z}}_{i-1|i-1} \tag{11a}$$

$$\boldsymbol{P}_{i|i-1} = \boldsymbol{F}_i \boldsymbol{P}_{i-1|i-1} \boldsymbol{F}_i^{\mathrm{T}} . \tag{11b}$$

During the update step, the filter computes

$$y_i = \hat{x}_i - \boldsymbol{h}_i \tilde{\boldsymbol{z}}_{i|i-1} \tag{11c}$$

$$\boldsymbol{C}_i = \boldsymbol{h}_i \boldsymbol{P}_{i|i-1} \boldsymbol{h}_i^{\mathrm{T}} + \sigma_i \tag{11d}$$

$$\boldsymbol{G}_i = \boldsymbol{P}_{i|i-1} \boldsymbol{h}_i^{\mathrm{T}} \boldsymbol{C}_i^{-1} \tag{11e}$$

$$\hat{\boldsymbol{z}}_{i|i} = \hat{\boldsymbol{z}}_{i|i-1} + \boldsymbol{G}_i y_i \tag{11f}$$

$$\boldsymbol{P}_{i|i} = (\boldsymbol{I}_2 - \boldsymbol{G}_i \boldsymbol{h}_i) \boldsymbol{P}_{i|i-1} , \tag{11g}$$

where $\hat{\boldsymbol{z}}_{i|i}$ and $\boldsymbol{P}_{i|i}$ are the *updated a posteriori* state estimate and its covariance, respectively, while $\boldsymbol{G}_i$ is the Kalman gain. The prediction error $\boldsymbol{y}_i$ is called *innovation* of the Kalman filter; together with its covariance $\boldsymbol{C}_i$, the innovation is exploited to compute

$$\beta_n = y_i^{\mathrm{T}} \boldsymbol{C}_i^{-1} y_i , \tag{12}$$

which Bob uses as an input for authenticity verification. We remark that, different from the general model of the Kalman filter, we have no control input. A more detailed description of the Kalman filter can be found in [19].

The considered feature yields the linear relations (8) and (10): in general, by choosing a different set of features, these relations may not be linear anymore; in this latter cases it becomes necessary to resort to the extended Kalman filter (EKF).

4.3 Authenticity Verification

In this section we propose several possible forms of the classification function $g(\cdot)$, which Bob uses to verify the authenticity of a packet. We focus on one-class classification solutions, i.e., $g(\cdot)$ can be designed and trained by using only observations from transmissions by Alice. To the best of the authors' knowledge, there is no optimal test for one-class classification, except in specific contexts [37]. Thus, we investigate three classification functions: a) a function based on the linear combination (LC) of the inputs; b) a classifier using an autoencoder (AE) neural network (NN); and c) a classifier based on a one-class support vector machine (OC-SVM).

Linear Combination (LC). The first classifier involves the linear combination of the entries of vector $\boldsymbol{\beta}$. In more detail, considering that Bob collects K innovations for each of the N receivers, he combines them as

$$g_{\mathrm{LC}}(\boldsymbol{\beta}) = \sum_{n=1}^{N} \sum_{k=1}^{K} \alpha_{n,k} \beta_{n,k} . \tag{13}$$

Several strategies may be used to estimate the weights: for instance, in [9] the authors take into account the relative distance between each pair of sensors and the (estimated) distance between each sensor S_n and Alice. Here, we consider a worst case analysis where Bob equally weighs each term of $\boldsymbol{\beta}$, i.e., $\alpha_{n,k} = 1, \forall k = 1, \ldots, K$ and $\forall n = 1, \ldots, N$.

Autoencoder NNs (AEs). AEs are unsupervised feed-forward neural networks, whose task is to replicate the input to the output. In more detail, they can be decomposed into two subnets, called respectively *encoder*, f_{enc} and *decoder*, f_{dec}.

The task of the encoder NN is to project the input to a (typically) lower-dimensional space, called the *latent space*. Thus, the encoder associates the input to its representation in the latent space. Conversely, the decoder NN reconstructs the input starting from its compressed version. In more detail, considering a training set of L legitimate sample vectors, $\{\boldsymbol{\beta}_\ell\}$, the AE is trained by minimizing the mean square error (MSE) training loss, i.e., by minimizing the reconstruction error

$$\min \frac{1}{L} \Gamma(\boldsymbol{\beta}_\ell) = \min \frac{1}{L} \sum_{\ell=1}^{L} \left\| \boldsymbol{\beta}_\ell - f_{\mathrm{dec}}\big(f_{\mathrm{enc}}(\boldsymbol{\beta}_\ell)\big) \right\|^2 . \tag{14}$$

More details about the AE design can be found in [14].

Typically, the latent space size is smaller than those of the input and output vectors, hence the reconstruction process is lossy. Still, the AE is trained to minimize the MSE loss, and is supposed to learn useful statistical properties of the training dataset. Thus, AEs can be used for authentication: if we train the AE by using only samples β_ℓ computed after Alice transmissions, only samples with the same statistical distribution are supposed to be reconstructed with low MSE [5, 28]. Therefore, the classifier will be $g_{\mathrm{AE}}(\boldsymbol{\beta}_\ell) = \Gamma(\boldsymbol{\beta}_\ell)$.

One-class SVM (OC-SVM). The idea behind an OC-SVM is to find the boundary that best encloses the training dataset samples. Next, during the testing phase, we will consider as legitimate only the samples falling within the boundary described by the SVM model.

In particular, considering a training dataset of size L, the testing function will be

$$g_{\text{SVM}}(\boldsymbol{\beta}) = \boldsymbol{\alpha}^{\text{T}}\phi(\boldsymbol{\beta}) + b\,, \tag{15}$$

and $\boldsymbol{\alpha}$ and b are respectively weights and bias of the trained OC-SVM classifier and $\phi(\cdot)$ is a suitable feature transformation function.

To train the classifier we consider the least squares SVM (LS-SVM) approach described in [7], where the loss function to be minimized is

$$\begin{aligned} &\min_{\boldsymbol{\alpha},b} \frac{1}{2}\boldsymbol{\alpha}^{\text{T}}\boldsymbol{\alpha} + b + C\frac{1}{2}\sum_{\ell=1}^{L} e_\ell^2\,, \\ &\text{with } e_\ell = -\boldsymbol{\alpha}^T\phi(\boldsymbol{\beta}_\ell) - b \quad \ell = 1,\ldots,L\,, \end{aligned} \tag{16}$$

where C is a hyper-parameter that has to be tuned depending on the training dataset itself [15].

5 NUMERICAL RESULTS

We evaluate our approach by simulating underwater acoustic communication channels via Bellhop [27]. We consider an operational region of about 6 km × 6 km, located in the San Diego bay area, having a depth between 250 and 600 m. The bottom-left corner of the area is located at coordinates (32°52′34.5″N, 117°24′12.8″W).

We deploy both Eve and Bob at random. Bob incorporates four sensors arranged as a tetrahedral pyramid of base radius and height equal to 5 m.

Alice moves across the area according to a correlated Gauss-Markov mobility model, and sends an acoustic signal once every Δt seconds. Specifically, Alice starts at a random location, $\boldsymbol{P}_{\text{A},0}$, with an initial velocity vector of magnitude $v_0 = \|\boldsymbol{v}_{\text{A},0}\|$ and direction drawn uniformly at random in an interval of 45° around due north. Once every $\Delta t = 1$ s, Alice's location $\boldsymbol{P}_{\text{A},i}$ and velocity $\boldsymbol{v}_{\text{A},i}$ are updated from step t_{i-1} to t_i, as

$$\boldsymbol{P}_{\text{A},i} = \boldsymbol{P}_{\text{A},i-1} + \boldsymbol{v}_{\text{A},i}\Delta t, \tag{17}$$

$$\boldsymbol{v}_{\text{A},i} = \alpha\,\boldsymbol{v}_{\text{A},i-1} + \boldsymbol{\eta}\sqrt{1-\alpha^2}\,, \tag{18}$$

where i and $i-1$ refer to the current and previous location and velocity update epochs, respectively, $\alpha = 1 - 2\cdot 10^{-3}$ is the trajectory correlation factor, and $\boldsymbol{\eta}$ is a Gaussian noise vector having (fixed) independent components of standard deviation $[2, 2, 1]$ m/s along the east–west, north–south and depth dimensions, respectively. These choices lead to correlated trajectories, which reproduce the uncertainty of drifting due to currents and eddies. At the same time, the lower variance along the depth dimension signifies a typically more accurate depth-keeping capability.

For every signal Alice transmits, we run Bellhop to compute the channel impulse response perceived at each of Bob's sensors as well as at Eve. Moreover, we reproduce different levels of randomness in Eve's estimate of Alice's location by displacing Alice uniformly at random within a radius of either 50, 100, 200, or 400 m over the azimuthal plane, and within a depth of ±20 m from Alice's actual location. These values are representative of realistic errors obtained via localization schemes based on matched field processing [12]. For each random displacement, we recompute the channel impulse response at each of Bob's sensors. Eve's uncertainty on Alice's location then translates into an uncertainty on the channel that Eve should reproduce in order to successfully impersonate Alice.

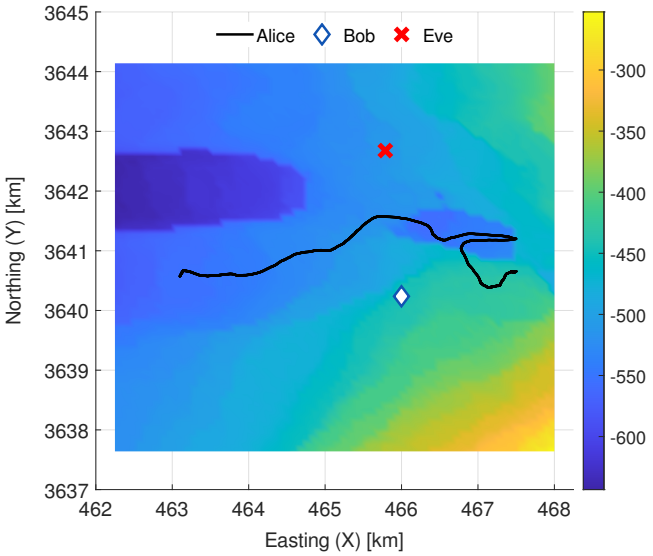

Figure 1: Example of simulation scenario showing the location of Eve and Bob, and a sample trajectory for Alice. The background colors convey the local depth.

Fig. 1 shows the bathymetry map of the area, the locations of Bob and Eve, and Alice's trajectory for one instance of our simulations. The full Monte-Carlo simulation set includes several realizations of the above scenario, with different locations and trajectories, as well as different movement speeds for Alice, i.e., where the initial velocity magnitude is $v_0 = 0.5, 1$, and 1.5 m/s. In total, we generated 20 simulations for each initial velocity magnitude v_0; each simulation lasts 12 000 s, corresponding to a total of 12 000 power-delay profiles collected per simulation.

We assume that, for each simulation, there is an initial training period when each sensor S_n receives data only from Alice and that each receiver will have such training dataset at her disposal. We remark that this training dataset has to be used to train both the (local) Kalman filter and the chosen function $g(\cdot)$. In particular, we considered a scenario where each sensor collected 600 (legitimate) feature vectors: thus, we used the first 200 power delay profiles, to train the Kalman filter; next, we input the latter 400 measurements to the Kalman filter, and extract the innovations that will be collected by Bob; this allows us to build the training dataset $\{\beta_\ell\}$ with $L = 400$ observations that will be used to train the actual $g(\cdot)$ functions. Notice that the training dataset size does not depend on K, i.e, the number of innovations that Bob collects from each sensor before making an authentication decision. Thus, by increasing K, we increase the size of each collected vector $\boldsymbol{\beta}$, but also shorten the training dataset. The remaining part of the legitimate power-delay profiles is used to compute the output in the legitimate case. Once both the Kalman filter and the $g(\cdot)$ function have been trained, we give up to K subsequent impulse responses associated to transmissions from Eve as input to each sensor.

For the Kalman filter, we set the state evolution noise covariance (8) to $\boldsymbol{Q}_i = 10^{-3}\,\boldsymbol{I}_2$ and the measurement noise variance (10) to $\sigma_i = 10^{-3}$. Moreover, we consider a worst-case scenario where each sensor has no information about Alice. Therefore we always

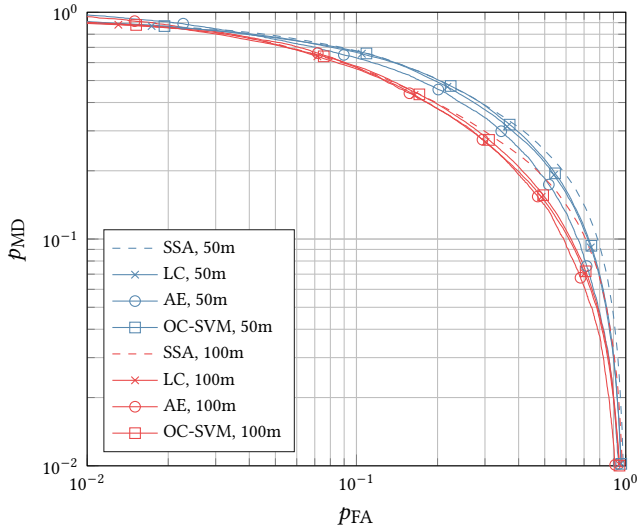

Figure 2: p_{MD} vs. p_{FA} for single-sensor authentication (SSA) and the described authentication verification functions, for a maximum localization error of 50 or 100 m over the azimuthal plane. $K = 1$. Autoencoder (AE): circle; linear combination (LC): cross; one-class support vector machine (OC-SVM): square.

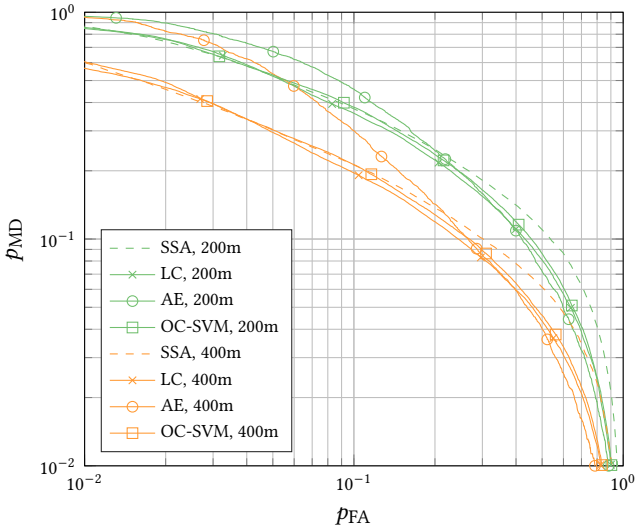

Figure 3: p_{MD} vs. p_{FA} for SSA and the described authentication verification functions, for a maximum localization error of 200 or 400 m over the azimuthal plane. $K = 1$. AE: circle; LC: cross; OC-SVM: square.

set the initial state to $\boldsymbol{x}_0 = [0, 0]^{\mathrm{T}}$. Indeed, having an initial (even partial) guess on Alice's distance and velocity would allow the Kalman filter to converge with a shorter training dataset, and leave more data to train the $g(\cdot)$ function.

As pointed out in Section 4.3, we assume that each sensor has no information about the others, thus for the LC approach (13) we set $\alpha_{n,k} = 1$. For the AE, following the results of [32], we designed both the encoder and the decoder to have one layer each, containing KN neurons. The size of the hidden layer is 2, since it provides the best classifier among the tested configurations. All the neurons have a linear activation function. The training lasted for 5 epochs. Finally, for the OC-SVM, we used a linear kernel function since it achieved better results than both the radial basis function and the polynomial kernels.

5.1 Performance Results

To evaluate the performance of our scheme, we consider the receiver operating characteristic (ROC) curves, obtained by plotting the FA and MD probabilities for different threshold values λ. For comparison, we consider also a single-sensor authentication (SSA) classifier, where Bob decides only based on the observation from his topmost sensor. Figs. 2 and 3 show the results for Alice's initial velocity $v_0 = 1\,\mathrm{m/s}$ and $K = 1$, using the LC, AE, OC-SVM and SSA classifiers, for different attacker estimation accuracies (50 m and 100 m in Fig. 2, 200 m and 400 m in Fig. 3).

As expected, if Eve can estimate the location of Alice more accurately, she has higher chances of successfully impersonating Alice: in other words, if we fix a given p_{FA}, p_{MD} becomes increasingly higher when the estimate of Alice's location becomes increasingly accurate. Instead, by comparing the different implementations of the function $g(\cdot)$, we notice that all the approaches outperform the SSA, meaning that all proposed schemes can successfully merge local information from different sensors. Moreover, in critical scenarios where Eve's estimate of Alice's location is most accurate, there exist negligible performance differences among the approaches. Conversely, for higher position estimation errors, the *AE* achieves the worst performance, while the LC and the OC-SVM methods are almost equivalent, with a slight edge for the LC. This may hint to the fact that the components of the vector $\boldsymbol{\beta}$ are (at least close to be) statistically independent.

Figs. 4 and 5 show the results for the same settings and algorithms, but Bob now collects $K = 3$ observation from each sensor into vector $\boldsymbol{\beta}$. This setting improves the classification performance.[3] For example, by comparing Fig. 5 to Fig. 3, we observe that setting the threshold λ to achieve a progressively lower p_{MD} leads to a much slower increase in p_{FA} (e.g., $p_{\mathrm{FA}} = 0.06$ for $p_{\mathrm{MD}} = 0.1$ if $K = 3$, against $p_{\mathrm{FA}} = 0.25$ for $K = 1$). We still observe that the AE classifier achieves the worst performance, whereas the LC and the OC-SVM are practically equivalent.

Finally, we investigate how a different average movement speed for Alice affects the classification performance. Besides $v_0 = 1\,\mathrm{m/s}$ as in the previous results, we now considers also $v_0 = 0.5$ and $v_0 = 1.5\,\mathrm{m/s}$. The corresponding results are shown in Figs. 6 and 7, respectively for $K = 1$ and $K = 3$. Because the LC classifier exhibits the best tradeoff between complexity and classification performance, we consider only LC in these result, and assume that Eve's accuracy in estimating Alice's location is 200 m. For $K = 1$, we observe no significant difference between the performance of LC for different speeds. However, increasing K (besides leading to better performance for the LC against the SSA classifier) leads to

[3] We tested several options for K, and found that $K > 3$ yields negligibly better results with respect to $K = 3$. The corresponding results are omitted for brevity.

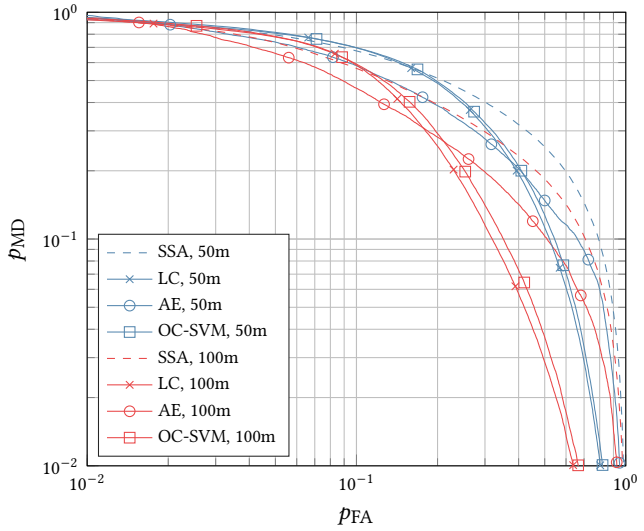

Figure 4: p_{MD} vs. p_{FA} for SSA and the described authentication verification functions, for a maximum localization error of 50 or 100 m over the azimuthal plane. $K = 3$.

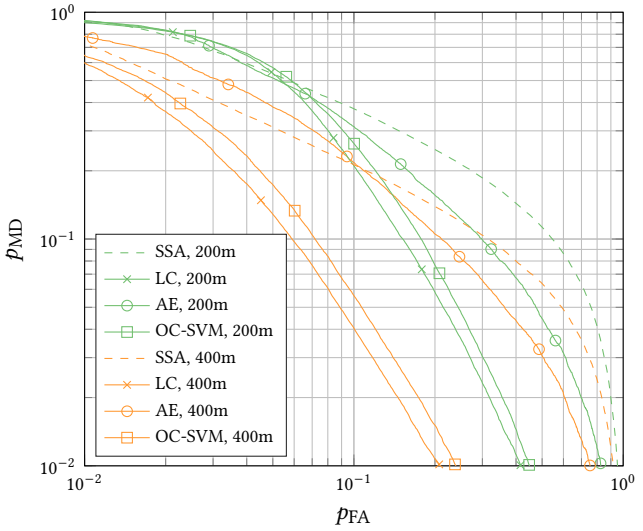

Figure 5: p_{MD} vs. p_{FA} for SSA and the described authentication verification functions, for a maximum localization error of 200 or 400 m over the azimuthal plane. $K = 3$.

an interesting outcome: the best results are obtained for $v_0 = 1\,\text{m/s}$. This suggests that there are two competing ways in which speed affects the power-weighed average delay metric we use for authentication. On the one hand, a lower speed implies that channels are more coherent over time, so that Eve has good chances to impersonate Alice successfully, even if her estimate of Alice's location is not too accurate. On the other hand, for $v_0 = 1.5\,\text{m/s}$, the average delay metric tends to change more abruptly, and the innovations computed by the Kalman filter becomes higher with each new legitimate transmission. This also translates into an advantage for Eve, as she decreases the margin the classifier needs to tell apart legitimate and impersonating transmissions.

6 CONCLUSIONS

We discussed an authentication algorithm for underwater acoustic networks with a mobile transmitter. We take a physical-layer security approach, and use a Kalman filter to establish whether the power-averaged delay of significant acoustic channel taps remains similar across subsequent received transmissions or not. We make the final decision by fusing the innovation observations from different Kalman filters, that process the data of different co-located acoustic receivers. For the fusion itself, we consider a linear combiner, an autoencoder, and a one-class support vector machine.

After evaluating the tradeoff between the false alarm and the missed detection probability for different values of the decision threshold, we show that fusing multiple observations over time improves the authentication performance, and that a simple linear combiner compares well against more complex observation fusion algorithms.

Future work will include distributing Bob's sensors across the network area (thus requiring them to wirelessly communicate their decisions to a central entity, so that Eve can attempt to interfere with the process); testing the performance of our approach when all of Alice, Bob and Eve move; exploiting different channel features in addition to the average tap delay; and exploring the impact of imperfect synchronization.

ACKNOWLEDGEMENTS

This work was sponsored in part by the NATO Science for Peace and Security Programme under grant no. G5884 (SAFE-UComm), and by the Italian Ministry of Economy and Finance through iNEST (Interconnected NordEst Innovation Ecosystem), funded by PNRR (Mission 4.2, Investment 1.5), Next Generation EU (Project ID: ECS 00000043, Digital, Industry, Aerospace).

REFERENCES

[1] Ahmed Al Guqhaiman, Oluwatobi Akanbi, Amer Aljaedi, and C. Edward Chow. 2020. Lightweight Multi-factor Authentication for Underwater Wireless Sensor Networks. In *Proc. CSCI*. IEEE, Las Vegas, NV, USA, 188–194.

[2] Waqas Aman, Zeeshan Haider, S. Waqas H. Shah, M. Mahboob Ur Rahman, and Octavia A. Dobre. 2020. On the Effective Capacity of an Underwater Acoustic Channel under Impersonation Attack. In *Proc. IEEE ICC*. IEEE, Dublin, Ireland.

[3] F. Ardizzon, S. Tomasin, R. Diamant, and P. Casari. 2022. Machine Learning-Based Distributed Authentication of UWAN Nodes With Limited Shared Information. In *Proc. UCOMMS*. Lerici, Italy.

[4] L. Bragagnolo, F. Ardizzon, N. Laurenti, P. Casari, R. Diamant, and S. Tomasin. 2021. Authentication of Underwater Acoustic Transmissions via Machine Learning Techniques. In *Proc. IEEE COMCAS*. IEEE, Tel Aviv, Israel, 255–260.

[5] Alessandro Brighente, Francesco Formaggio, Giorgio Maria Di Nunzio, and Stefano Tomasin. 2019. Machine Learning for In-Region Location Verification in Wireless Networks. *IEEE Journal on Selected Areas in Communications* 37, 11 (2019), 2490–2502.

[6] Zhongyue Chen, Huifang Chen, and Wen Xu. 2014. Simplified time synchronization for underwater acoustic sensor networks with high propagation latency. In *Proc. MTS/IEEE OCEANS*. IEEE, Taipei, Taiwan, 1–5.

[7] Young-Sik Choi. 2009. Least Squares One-Class Support Vector Machine. *Pattern Recogn. Lett.* 30, 13 (oct 2009), 1236–1240.

[8] R. Diamant. 2016. Closed Form Analysis of the Normalized Matched Filter With a Test Case for Detection of Underwater Acoustic Signals. *IEEE Access* 4 (2016), 8225–8235.

[9] Roee Diamant, Paolo Casari, and Stefano Tomasin. 2019. Cooperative Authentication in Underwater Acoustic Sensor Networks. *IEEE Trans. Wireless Commun.* 18, 2 (2019), 954–968.

[10] Roee Diamant and Lutz Lampe. 2013. Underwater Localization with Time-Synchronization and Propagation Speed Uncertainties. *IEEE Trans. Mobile Comput.* 12, 7 (2013), 1257–1269.

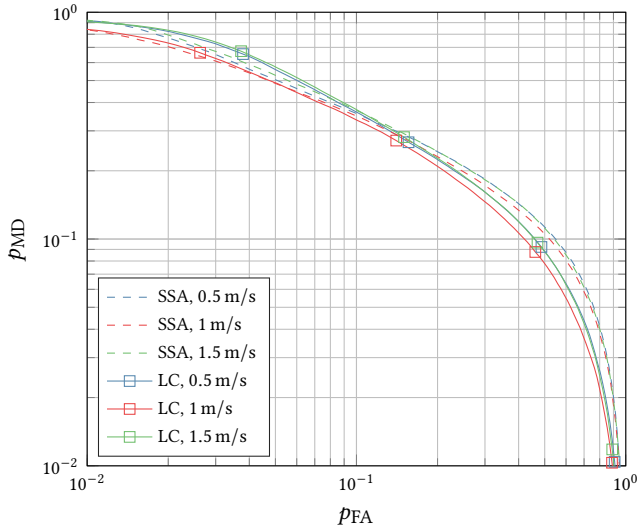

Figure 6: p_{MD} vs. p_{FA} for SSA and the described authentication verification functions, for a maximum localization error of 200 m over the azimuthal plane and $v_0 = 0.5$, 1, or 1.5 m/s. $K = 1$.

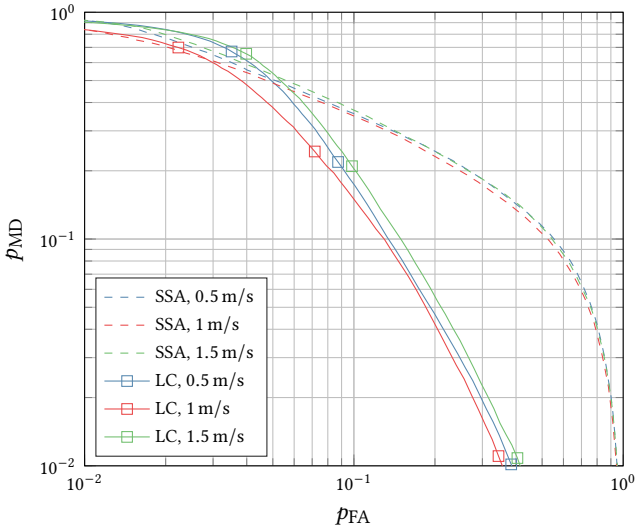

Figure 7: p_{MD} vs. p_{FA} for SSA and the described authentication verification functions, for a maximum localization error of 200 m over the azimuthal plane and $v_0 = 0.5$, 1, or 1.5 m/s. $K = 3$.

[11] Gianluca Dini and Angelica Lo Duca. 2012. A Secure Communication Suite for Underwater Acoustic Sensor Networks. *MDPI Sensors* 12 (2012), 15133–15158. http://dx.doi.org/10.3390/s121115133

[12] E. Dubrovinskaya, P. Casari, and R. Diamant. 2019. Bathymetry-aided Underwater Acoustic Localization using a Single Passive Receiver. *J. Acoustic Soc. Am.* 146, 6 (Dec. 2019), 4774–4789.

[13] E. Dubrovinskaya, V. Kebkal, O. Kebkal, K. Kebkal, and P. Casari. 2020. Underwater Localization via Wideband Direction-of-Arrival Estimation Using Acoustic Arrays of Arbitrary Shape. *Sensors, Special Issue "Internet of Underwater Things"* 20, 14 (July 2020), 1–20.

[14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning.* MIT Press, Cambridge, MA, USA. http://www.deeplearningbook.org.

[15] X.C. Guo, J.H. Yang, C.G. Wu, C.Y. Wang, and Y.C. Liang. 2008. A novel LS-SVMs hyper-parameter selection based on particle swarm optimization. *Neurocomputing* 71, 16 (Oct. 2008), 3211–3215. https://www.sciencedirect.com/science/article/pii/S0925231208002932

[16] Al Amin Islam and Kazi Abu Taher. 2021. A Novel Authentication Mechanism for Securing Underwater Wireless Sensors from Sybil Attack. In *Proc. ICEEICT.* IEEE, Mirpur, Dhaka, 1–6.

[17] Shengming Jiang. 2019. On Securing Underwater Acoustic Networks: A Survey. *IEEE Commun. Surveys Tuts.* 21, 1 (2019), 729–752.

[18] E. Jorswieck, S. Tomasin, and A. Sezgin. 2015. Broadcasting Into the Uncertainty: Authentication and Confidentiality by Physical-Layer Processing. *Proc. IEEE* 103, 10 (Oct. 2015), 1702–1724.

[19] S. Kay. 1993. *Fundamentals of Statistical Signal Processing: Estimation Theory.* Prentice-Hall, Englewood Cliffs, NJ.

[20] Muhammad Khalid, Ruiqin Zhao, and Nauman Ahmed. 2020. Physical Layer Authentication in Line-of-Sight Underwater Acoustic Sensor Networks. In *Proc. MTS/IEEE OCEANS.* IEEE, Singapore, 1–5.

[21] C. Lal, R. Petroccia, K. Pelekanakis, M. Conti, and J. Alves. 2017. Toward the Development of Secure Underwater Acoustic Networks. *IEEE J. Ocean. Eng.* 42, 4 (Oct. 2017), 1075–1087.

[22] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. 1996. *Handbook of Applied Cryptography.* CRC Press, Boca Raton, FL.

[23] Zoi-Heleni Michalopoulou, Peter Gerstoft, and Diego Caviedes-Nozal. 2021. Matched field source localization with Gaussian processes. *JASA Express Lett.* 1, 6 (2021).

[24] Dalhatu Muhammed, Mohammad Hossein Anisi, Mahdi Zareei, and Anwar Khan. 2018. Game Theory-Based Cooperation for Underwater Acoustic Sensor Networks: Taxonomy, Review, Research Challenges and Directions. *Sensors* 18, 2 (Feb. 2018). art. 425.

[25] K. Pelekanakis, C. M. G. Gussen, R. Petroccia, and J. Alves. 2019. Robust Channel Parameters for Crypto Key Generation in Underwater Acoustic Systems. In *Proc. MTS/IEEE OCEANS.* IEEE, Seattle, WA, USA, 1–7.

[26] K. Pelekanakis, S.A. Yıldırım, G. Sklivanitis, R. Petroccia, J. Alves, and D. Pados. 2021. Physical Layer Security against an Informed Eavesdropper in Underwater Acoustic Channels: Feature Extraction and Quantization. In *Proc. UCOMMS.* IEEE, 1–5.

[27] M. Porter et al. 2018. Bellhop code. Retrieved Aug. 31, 2022 from http://oalib.hlsresearch.com/Rays/index.html

[28] Manassés Ribeiro, André Eugênio Lazzaretti, and Heitor Silvério Lopes. 2018. A study of deep convolutional auto-encoders for anomaly detection in videos. *Pattern Recognition Letters* 105, C (apr 2018), 13–22.

[29] Mohammad Sharif-Yazd, Mohammad Reza Khosrav, and Mohammad Kazem Moghimi. 2017. A Survey on Underwater Acoustic Sensor Networks: Perspectives on Protocol Design for Signaling, MAC and Routing. *SciRP J. of Computer and Commun.* 5, 5 (Feb. 2017), 12–23.

[30] G. Sklivanitis, K. Pelekanakis, S.A. Yıldırım, R. Petroccia, J. Alves, and D. Pados. 2021. Physical Layer Security against an Informed Eavesdropper in Underwater Acoustic Channels: Reconciliation and Privacy Amplification. In *Proc. UCOMMS.* IEEE, 1–5.

[31] Evaldo Souza, Hao Chi Wong, Italo Cunha, Ítalo Cunha, L. F. M. Vieira, and Leonardo B. Oliveira. 2013. End-to-end authentication in Under-Water Sensor Networks. In *Proc. IEEE ISCC.* IEEE, Split, Croatia, 299–304.

[32] Harald Steck and Dario Garcia Garcia. 2021. On the Regularization of Autoencoders. https://arxiv.org/abs/2110.11402

[33] M. Stojanovic and J. Preisig. 2009. Underwater acoustic communication channels: Propagation models and statistical characterization. *IEEE Commun. Mag.* 47, 1 (2009), 84–89.

[34] Inam Ullah, Jingyi Chen, Xin Su, Christian Esposito, and Chang Choi. 2019. Localization and Detection of Targets in Underwater Wireless Sensor Using Distance and Angle Based Algorithms. *IEEE Access* 7 (2019), 45693–45704.

[35] Arjan Vermeij and Andrea Munafò. 2015. A Robust, Opportunistic Clock Synchronization Algorithm for Ad Hoc Underwater Acoustic Networks. *IEEE J. Ocean. Eng.* 40, 4 (2015), 841–852.

[36] Chi Yuan, Wenping Chen, Yuqing Zhu, Deying Li, and Jie Tan. 2015. A Low Computational Complexity Authentication Scheme in Underwater Wireless Sensor Network. In *Proc. MSN.* IEEE, Shenzhen, China, 116–123.

[37] O. Zeitouni, J. Ziv, and N. Merhav. 1992. When is the generalized likelihood ratio test optimal? *IEEE Trans. Inf. Theory* 38, 5 (1992), 1597–1602.

[38] Shuailiang Zhang, Xiujuan Du, and Xin Liu. 2020. A Secure Remote Mutual Authentication Scheme Based on Chaotic Map for Underwater Acoustic Networks. *IEEE Access* 8 (2020), 48285–48298.

[39] Ruiqin Zhao, Muhammad Khalid, Octavia A. Dobre, and Xin Wang. 2022. Physical Layer Node Authentication in Underwater Acoustic Sensor Networks Using Time-Reversal. *IEEE Sensors J.* 22, 4 (2022), 3796–3809.