Secret Key Generation from Route Propagation Delays for Underwater Acoustic Networks

Roee Diamant, *Senior Member, IEEE*, Stefano Tomasin, *Senior Member, IEEE*,
Francesco Ardizzon, *Student Member, IEEE*, Davide Eccher, and Paolo Casari, *Senior Member, IEEE*

***Abstract*—With the growing use of underwater acoustic communications and the recent adoption of standards in this field, it is becoming increasingly important to secure messages against eavesdroppers. In this paper, we focus on a physical-layer security solution to generate sequences of random bits (keys) between two devices (Alice and Bob) belonging to an underwater acoustic network (UWAN); the key must remain secret to a passive eavesdropper (Eve) not belonging to the UWAN. Our method is based on measuring the propagation delay of the underwater acoustic channel over multiple hops of the UWAN: this harvests the randomness in the UWAN topology and turns the slow sound propagation in water into an advantage against eavesdropping. Our key generation protocol includes a route discovery handshake, whereby all UWAN devices at intermediate hops accumulate their message processing delays. This enables Alice and Bob to compute the actual propagation delays along each route and to map such information to a sequence of bits. Finally, from these bit sequences, Alice and Bob obtain a secret key. We analyze the performance of the protocol theoretically and assess it via extensive simulations and field experiments.**

***Index Terms*—Underwater acoustic networks; physical layer security; secret-key generation; time of flight; simulation; Bellhop; lake experiments**

I. Introduction

Advances in underwater acoustic communications and the decreasing cost of acoustic sensors are progressively turning underwater acoustic networks (UWANs) into feasible tools for undersea operations such as seabed monitoring, contamination control, and search-and-survey operations. These applications require multiple cooperative submerged sensors to communicate with one another. When defense-related or mission-critical communications are involved (e.g., with devices monitoring marine infrastructures such as oil and gas rigs), ensuring secure communications is a fundamental prerequisite.

R. Diamant (roee.d@univ.haifa.ac.il) is with the Department of Marine Technologies, University of Haifa, 3498838 Haifa, Israel.

S. Tomasin (tomasin@dei.unipd.it) and F. Ardizzon are with the Department of Information Engineering, University of Padova, 35131 Padova, Italy.

D. Eccher and P. Casari (paolo.casari@unitn.it) are with DISI, University of Trento, 38123 Trento, Italy. S. Tomasin, F. Ardizzon, D. Eccher, and P. Casari are also with Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT), 43124 Parma, Italy.

This work was sponsored in part by the NATO Science for Peace and Security Programme under grant no. G5884 (SAFE-UComm), and from 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).

The authors extend their gratitude to the team of the CUS nautical center of the University of Trento for their great logistic support, to Gabriele Stulzer for his assistance during the execution of the experiments, and to the team of Witted (Rovereto, Italy) for their help with the first in-water tests.

The foundation of many security solutions based on cryptography is the availability of some secret information (called the *key*), shared by two legitimate parties but unknown to eavesdropping devices. The key must be refreshed from time to time to prevent attacks based on the long-term observation of encrypted messages exchanged by legitimate nodes. One option is to pre-share keys across all devices before each UWAN deployment, or equivalently, install a pre-designed cypher module on all nodes. This solution has three important disadvantages: (a) it may entail long delays in multi-party deployments, as it requires to get all devices together in the same place to install the key, or to establish a secure off-band channel for key sharing; (b) if an attacker successfully captures a node, it will be able to reveal the key and consequently tap on all communications; and (c) any secret key must be refreshed after several uses, which would require new installs.

A more compelling option is to continuously refresh secret keys via physical-layer security approaches. With this method, appropriate features of the underwater acoustic communication channels provide a source of randomness, and suitable protocols based on information-theoretic security ensure the confidentiality of the generated key against an eavesdropper. While physical-layer secret key generation has been well investigated for terrestrial communications, the unique characteristics of the acoustic channel make it hard to readily port the same solutions to the underwater acoustic context [1], [2], as discussed in more detail in Section II.

In this paper, we propose a secret key generation procedure based on the propagation delay properties of underwater acoustic channels. Once accumulated throughout multiple hops of a UWAN, the delay measurement harvests the UWAN topology's randomness, and turns the slow sound propagation speed (roughly 1500 m/s) and correspondingly large times of flight (ToFs) of acoustic transmissions into an advantage against eavesdropping. In our approach, a two-way packet exchange between Alice and Bob allows both ends to compute and quantize these ToF values as packets travel along different routes. This information is partially secret to Eve: hence, we can extract a subset of bits shared by Alice and Bob and secret to Eve using information-theoretic solutions for secret key agreement [3, Ch. 4]. Moreover, the sparsity of typical UWAN topologies, resulting from the significant acoustic power attenuation under water, typically limits Eve to hear only a fraction of the data transmitted by legitimate nodes.

Our solution introduces a minimal communication overhead thanks to a simple message exchange between Alice and Bob, while the key is generated in a non-federated fashion.

By way of contrast, applying full-fledged security protocol suites such as [4] would require to change the format of the transmitted data, and thus limits portability over existing modems. Moreover, our solution is based on the assumption that the propagation delay over each of the network's links is a property of each link, and cannot be easily obtained by Eve. A support to this argument is that, even when applying powerful graph localization algorithms (e.g., [5]), Eve cannot fully reconstruct the propagation delays in the network, unless she is exposed to all transmissions. We note that our solution does not contain any concealed information, and that all knowledge available to Alice and Bob is public and assumed to be known by Eve. With respect to existing secret-key generation solutions, the main contributions of our paper are the following:

1) we consider the propagation delay of the underwater acoustic channel and the network topology as sources of randomness;
2) we propose a secret-key agreement protocol operating *across nodes* in the entire UWAN and exploiting the random topology of the UWAN;
3) we assess the performance of the proposed technique by both simulation and seven field experiments.

The remainder of this paper is organized as follows. After describing related work (§II), we introduce our system model and assumptions (§III), detail our secret key generation approach (§IV), and discuss the main information-theoretic security metrics of our proposed scheme (§V). In §VI, we present simulation results, including realistic acoustic propagation modeling (§VI-B). We then discuss the results of seven field experiments (§VII), and draw our conclusions in §VIII.

II. Literature Background and Contribution

While the exploitation of noisy channels for secrecy was proposed by Wyner [6], secret-key agreement was later proposed by Maurer [7], Ahlswede and Csiszar [8]. In these works, the bounds on the achievable secret-key capacity had been proposed and proved. Secret-key agreement has then been applied in various contexts, especially on radio channels [9]–[12]. A detailed description of the main steps of the physical-layer key generation procedure, i.e., advantage distillation, information reconciliation, and privacy amplification, can be found in [3].

Considering underwater acoustic communications, several features of the underwater channel can be exploited for physical-layer security purposes, including resistance against jamming [13], [14], denial-of-service (DoS) [4], securing exposed links through key generation [15], data integrity for message authentication [16], and covert communications [17]. About secret-key agreement in underwater scenarios, [18] leverages the channel frequency response of an OFDM system for underwater communications; furthermore, two enhancements are proposed, one based on adaptively weighting the probe signals to increase the channel correlation, and a second one to introduce a block-sliced key verification procedure that compensates for fast channel dynamics and increases the key agreement probability.

An overview of alternatives for key extraction, information reconciliation, and privacy application is provided, in particular for its application in an underwater context in [19]. An underwater OFDM system has been considered in [20], which uses Bose-Chaudhuri-Hocquenghem (BCH) codes for information reconciliation; a proof-of-concept implementation of the proposed approach has been tested in a lake. The solution has been further investigated in [21], by introducing adaptive pilot signals to estimate and compensate for channel dynamics, and in [22] that uses turbo codes for information reconciliation. In [15], the channel feature for the secret key agreement is the received signal strength. Another solution based on a suitable multistage channel sounding protocol is discussed in [23] to deal with channel variations and large underwater acoustic propagation delays. Channel impulse response features, such as its norm, a smooth sparseness measure, and the root-mean-square delay spread are exploited for key generation in [24], and demonstrated through experimental results. The works in [25], [26] extend the above by pursuing all steps of key generation including reconciliation and privacy amplification, showing acceptable bit disagreement ratios. The work in [27] investigates multipath-based features for secret key generation, with experimental results collected in a shallow-water experiment off the coast of Portugal.

The challenges in fitting key agreement protocols developed for radio channels to the underwater acoustic domain are summarized in [28], where a comparison between three key exchange protocols shows the trade-off between the transmission latency and the required energy. To generate secret keys, [29] uses the α-order Renyi entropy, and chooses the multipath and Doppler effects as a source of randomness. To overcome the assumption of independent channels, a surface station works as the common source of randomness. Still, a hard channel reciprocity assumption remains. A similar use of channel features is presented in [27], where the features exploited as source of randomness are the magnitude of the effective multipath, the sparseness of the channel, the delay spread, and the delay of the multipath arrivals. However, results show that the coherence time of the channel poses a major challenge to achieve key agreement. To cope with continuous channel variations, the key-exchange packets may be split into multiple transmission rounds [23]. The approach of [30] first authenticates a sensor node and then applies the secret key. An authentication procedure is proposed in [31], wherein an agreed hash function generates the keys while harvesting randomness from the received signal waveform. Lastly, [26] proposes to use the channel features as a source of randomness for key generation, assuming the Alice-Bob link is significantly different than that of Alice-Eve or Bob-Eve. By considering several channel features and assuming large variations to each, 256 bits for the key are achieved. Small changes between the Alice-Bob and Bob-Alice links are managed through a Reed-Solomon code. However, when the channel's coherence time is short, it will be harder to obtain the same key at the devices.

A. Our Contribution

The above solutions offer techniques for secret key generation based on the features of the channel. While these features are sources of randomness, we observe that, due to the fast variations in the underwater acoustic channel, it is challenging to obtain an ad hoc agreement on the secret key between Alice and Bob based on such features [23], [27], [28]. With this observation, we conclude that, for key generation, a good source of randomness has the following properties:

1) spatial dependency, such that Eve cannot directly obtain the key from the channel;
2) slow time dependency, such that the channel feature changes in time while allowing Alice and Bob to reach an agreement when generating the key;
3) large value variability, to produce as many secret bits as possible.

In this work, we argue that the *propagation delay* and the *network topology* are randomness sources that meet the above requirements. Not to be confused with the channel's delay spread, the propagation delay is determined by the nodes' locations, which are considered unknown in this work. Moreover, we remark that the propagation delay values can vary significantly throughout the network, due to the low sound propagation speed, and may also be very different depending on the route that a packet traverses while being forwarded from Alice to Bob or vice-versa. This approach is a unique contribution of our paper.

III. System Model

We consider a UWAN where N devices communicate over acoustic channels [32]. Two of them, called Alice and Bob, seek to obtain a sequence of bits (*secret key*) that must remain unknown to a third device called Eve, which is external to the UWAN. As Alice and Bob are connected through a multihop network, communications require relaying through the UWAN. In order to generate the key, Alice and Bob need a shared source of randomness. This can be found in the location-specific and time-varying physical features of the channels among any node pair in the UWAN. In this work, we exploit the topology of the UWAN, that induces unknown and diverse propagation delays over each link, which in turn will be used for key generation. Our network, communications and attacker models are introduced in the following.

Network Model: We assume that the topology can take any form from fully connected to a chain topology, since it is determined by random factors not immediately related to communications, e.g., current drifts or network node mobility. We then consider the UWAN topology as random: the number of network devices and their locations are unknown to all devices (including Alice, Bob, and Eve). Also, the network topology (and thus the end-to-end propagation delays) are assumed to change slowly over time. Since the network's topology as well as the number of nodes and their mobility within the network are considered unknown, in our analysis we assume the nodes' positions are random and statistically independent.

In our system model, we allow the motion of nodes. Yet, we assume that, during the exchange of end-to-end messages between Alice and Eve, the network topology remains the same both for the network nodes and the possible attacker positions. This is because, in underwater communication, due to flow noise and Doppler shift, nodes are likely to limit their speed to a maximum of $5\,\mathrm{kn}$. In the practical case of a packet exchange between Alice and Eve that takes $20\,\mathrm{s}$, this speed would correspond to roughly $50\,\mathrm{m}$, which is much shorter than the expected distance among nodes (order of km per hop).

Communication Model: We also assume that a) the UWAN nodes can estimate the retransmission delay introduced by their modem;[1] b) at least Alice and Bob are time synchronized, e.g., as a result of running an underwater time synchronization scheme (e.g., [33], [34]), but a later discussion on how to relax this assumption follows; and c) an underlying link-level protocol ensures correct packet reception, e.g., via automatic repeat queries (ARQ), whose delay can be determined by each receiver. In particular, we consider underlying medium access control (MAC) and network protocols that combat packet loss through packet re-transmission and forwarding, respectively. For example, in our lake experiment (see Section VII), we use an underwater modem that has built-in ARQ and relaying mechanisms, whose impact on the end-to-end delivery delay is accounted for by reporting the accumulated delay until a successful transmission in the header of the transmitted packet. As for the routing protocol, we rely on a flooding mechanism where intermediate nodes forward any incoming packet unless already forwarded. Other routing protocols may also be adequate, under the condition that the reciprocity of the paths between Alice and Bob holds.

Eve is assumed to be an overhearing entity that can detect and receive incoming packets based on her location in the network and on the received signal quality. For a detailed model of Eve and her attack strategy, we refer the reader to Section V-A.

In our model, we assume a maximum time T_c for a packet to propagate from Alice to Bob. This value works as an upper bound, that considers the transmission range of the used modem, the maximum number of nodes in the network, and the maximum delay required for effective error control during the packet forwarding process.

IV. Topology-based Key Agreement for UWANs

For the secret key agreement protocol, we consider the following steps [3, §4.3]: a) *sequence generation*, by which two sequences of M bits are obtained by Alice and Bob, b) *information reconciliation*, by which, through coding techniques, Alice and Bob remove differences in their bit sequences; and c) *privacy amplification*, by which a shorter bit sequence that is secret to Eve is extracted (typically using hash functions). In this paper, we focus on step a), which will exploit specific features of UWAN communications. Instead, steps b) and c) operate on bit sequences, hence they are not specific to

[1] One possible method is to determine the local reception start time, decide on the packet forwarding delay according to some protocol, and retransmit exactly after such delay since the begin of packet reception. This is also the solution we employ in our lake experiment, see Section VII.

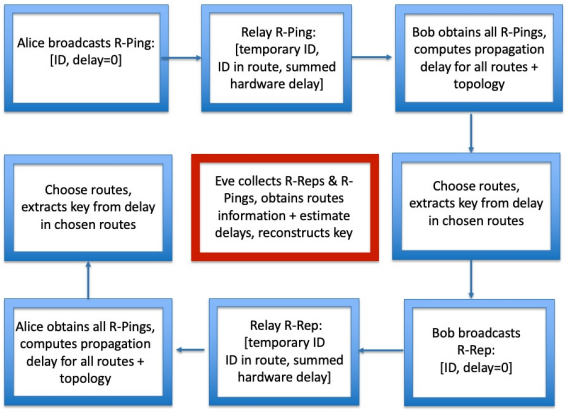


Fig. 1. Block diagram of the proposed algorithm.

UWANs and can be achieved through existing techniques, including approaches developed for UWAC scenarios [26]. For these reasons, b) and c) will not be part of our work.

In summary, the main process of our sequence generation mandates that Alice floods a Request-Ping (R-Ping) packet to Bob, and, in return, Bob floods a Request-Reply (R-Rep) packet to Alice. Both Alice and Bob estimate the per-route ToF as well as the network topology. The bit sequence used for the key extraction is generated separately at Alice and Bob by concatenating the ToFs of the N_c chosen routes, and by quantizing them down to a pre-determined quantization step ρ over the quantization interval $[0, T_c]$. The number of quantization bits per route is, therefore, $N_r = \lceil \log_2(T_c/\rho) \rceil$. Both T_c and N_r are considered to be public information. We also assume that the number of routes N_c along which Alice and Bob measure the propagation delay is also public. We now describe the sequence generation procedure in detail. Fig. 1 summarizes the procedure.

A. Route Delay Estimation

Referring to Fig. 1, the flooded R-Ping packet includes the ID of Alice, the transmission time T_s, and a field δ_i to store the accumulated hardware and software delay over the i-th route. Each node different from Alice and Bob that receives the R-Ping appends its local ID to the packet, updates δ_i, and forwards the packet only if the node does not find its own temporary ID already in the list of traversed relays. This prevents the appearance of loops in the routes. After receiving a packet at time T_r, Bob extracts the IDs of the nodes forming route i and the corresponding accumulated delay, δ_i. He then computes the *net route delay* as

$$D_i = T_r - T_s - \delta_i \ . \tag{1}$$

In the above process, we mitigate collisions and packet failures at the link level, e.g., via repeated transmissions or the ARQ policies mentioned in Section III. [2] In this case, nodes are required to measure the time elapsed between the reception of the packet and each re-transmission, and accumulate this as part of the hardware delay field of the packet, δ_i. Thanks to this delay accumulation step, Bob can obtain the propagation delay along the route through (1), and thus exclude any other contribution to the total delay. We note that our assumption of underlying ARQ handles the case of lost packets, while the inclusion of the delay of ARQ retransmissions in the accumulated path delay compensates for such delays towards key agreement. As such, while Eve may have a better receiver to correctly receive more packets than Alice or Bob, packets lost by Alice and Bob will eventually get to the later nodes. Thus the cost of such losses is only a delay in the procedure of the key exchange. We also note that this option is not available for Eve, who cannot request re-transmissions like Alice and Bob and is thus more sensitive to lost packets.

After receiving the first R-Ping, Bob keeps listening for a globally known time interval of duration T_c, to receive additional R-Pings. For the above process to be successful, R-Ping and R-Rep transmissions should be reliable, and T_c should be large enough for the collection of all R-Reps. We remark there exist practical limits to retransmission policies. For example, an exceeding number of R-Ping and R-Rep transmissions may inconveniently prolong the delay of the multihop flooding process, and may lead R-Reps to observe different topologies with respect to R-Pings. For this reason, it is important that Bob (Alice) keep listening for R-Pings (R-Reps) for a carefully chosen time interval T_c. In fact, T_c strikes a balance between topology coherence and the maximum route length: it should allow ample time for R-Pings to arrive and reveal the network topology to Bob. At the same time, it should not be exceedingly long, otherwise R-Pings and R-Reps end up observing different network topologies.

Besides choosing a proper value for T_c, a suitable MAC-level scheme can help in reducing or avoiding packet failures due to collisions. However, efficient flooding protocols typically focus on reducing the number of packet replicas. Relevant examples of this design are, e.g., Dflood [35] and MPR-Light [36]. Conversely, in our scheme we do need to map all available routes, and cannot resort to restricted flooding, because each relay modifies an R-Ping or R-Rep packet by appending its own ID to the list of traversed relays. Therefore, we mitigate collisions through carrier-sense multiple access (which helps avoid receiver deafness) and via random backoffs before each packet transmission (which further reduces the chance of collisions).

Moreover, we need to maximize the probability that, ideally, all neighbors of the source (Alice), of the sink (Bob) and of any intermediate relays overhear R-Ping and R-Rep packets, and thus have the chance to insert their own ID in them. This can be achieved, e.g., with repeated packet transmissions [37], rateless coding [38], or network coding solutions that would mix messages from different relays [39].

After T_c has passed, Bob reconstructs the topology matrix, and singles out a set of N_c routes,[3] $\mathcal{S} = \{D_1, \ldots, D_{N_c}\}$, through a method described later in Section IV-B. The delay

[2] Unrecovered packet failures may lead to the reconstruction of different routes by Alice and Bob, which will lead to different bit sequences. These differences can be in part fixed by the key reconciliation process.

[3] If N_c is larger than the number of actual routes, all the available routes are considered to extract the key.

D_i for each chosen route is quantized down to a propagation delay step ρ, and transformed into a sequence of N_r bits, where N_r depends on the maximum delay in the UWAN. To achieve agreement between Alice and Bob, the bit sequences obtained from all quantized delay values $\{\hat{D}_i\}$, $i \in \mathcal{S}$ are concatenated by their ToF order. Bob repeats the above process by sending a R-Rep message to Alice. The message propagates through the network subject to the same rules of the R-Ping, until Alice has also reconstructed the network topology, chosen the N_c routes, and obtained the sequence bits to be used for key generation. The process ends after $2T_c$.

B. Choice of the Routes

Alice and Bob decide on a set of N_c routes via the following process. Let $\widehat{\mathbf{A}}_{\text{Bob}}$ (respectively, $\widehat{\mathbf{A}}_{\text{Alice}}$) be the network connectivity matrix that Bob (respectively, Alice) reconstructs after receiving the R-Pings (respectively, R-Reps) flooded through the network. In these matrices, entry (i, j) is set to 1 if there is a communication link between nodes i and j. We recall that, while channel variability due to the doubly-spread nature of underwater acoustic channels as well as packet collisions may lead to $\widehat{\mathbf{A}}_{\text{Alice}} \neq \widehat{\mathbf{A}}_{\text{Bob}}$, it is still possible to use MAC-level schemes to correct for such errors, e.g., through repeated transmissions of R-Pings and R-Reps [37], or error-correcting codes [38], [39].

We focus now on the process at Bob, as it is entirely equivalent to the one at Alice. Say that Bob receives N_{Bob} R-Pings. From each of them, Bob retrieves the route that the R-Ping traversed. Let us denote this set of routes as $\mathcal{R} = \{R_1, R_2, \cdots, R_{N_{\text{Bob}}}\}$, and assume that each route is an ordered tuple containing the traversed nodes, i.e., $R_k = (n_0, n_1, n_2, \cdots, n_{L_k-1}, n_{L_k})$, where the route contains L_k hops, n_0 is Alice and n_{L_k} is Bob, without loss of generality. Moreover, denote a sub-route of R_k containing only nodes from ℓ to m as $R_k^{[\ell,m]}$.

The objective of the route selection process is, ideally, to single out disjoint routes that have a different number of hops. This caters for a) minimal relay reuse, otherwise Eve would gain a significant advantage by positioning itself close to a bottleneck relay, and b) route delay diversity, as longer routes necessarily take longer to traverse than shorter routes.

To achieve the above, Bob forms an auxiliary symmetric matrix $\mathbf{A}^{(R)} = (a_{ij}^{(R)})$ of size $N_{\text{Bob}} \times N_{\text{Bob}}$, where each entry $a_{ij}^{(R)}$ conveys the jointness of routes i and j as follows. Formally, call $\widehat{\mathbf{A}}_{\text{Bob}}\big(R_i^{[i_1,i_2]}, R_j^{[j_1,j_2]}\big)$ the sub-matrix obtained by selecting the rows of $\widehat{\mathbf{A}}_{\text{Bob}}$ indexed by the node indices of route R_i from position i_1 to position i_2, and the columns indexed by the node indices of route R_j from position j_1 to position j_2. We define

$$a_{ij}^{(R)} = \mathbb{1}\Big(\widehat{\mathbf{A}}_{\text{Bob}}\big(R_i^{[1,L_i-1]}, R_j^{[1,L_j-1]}\big)\Big), \tag{2}$$

where $\mathbb{1}(\cdot)$ counts the number of ones in the argument, and superscripts $[1, L_i-1]$ and $[1, L_j-1]$ mean that we exclude the indices of Alice and Bob from both routes. Eq. (2) is equivalent to evaluating the connectivity of the sub-network that includes all relays in routes R_j and R_j, except Alice and Bob. Recall that indices i and j refer to the index of the respective route in the list of routes collected by Bob. For example, if $R_i = (1, 4, 3, 6)$ and $R_j = (1, 2, 4, 5, 6)$, then Alice is node 1, Bob is node 6, $L_i = 4$, $L_j = 5$, and $\widehat{\mathbf{A}}_{\text{Bob}}\big(R_i^{[1,L_i-1]}, R_j^{[1,L_j-1]}\big)$ is the submatrix that includes rows 4 and 3, and columns 2, 4 and 5 of $\widehat{\mathbf{A}}_{\text{Bob}}$. Bob chooses the first route as

$$R_1^\star = \arg\min_k \ \widehat{\mathbf{A}}_{\text{Bob}}\, \mathbf{1}_{N_{\text{Bob}}\times 1}\,, \tag{3}$$

where $\mathbf{1}_{a\times b}$ is the matrix of all ones and size $a \times b$, and $1 \leq k \leq N_{\text{Bob}}$ spans the column of the matrix. This singles out the route (among those received in the R-Pings) that is most disjoint from all other routes, in the sense of the sub-network connectivity metric introduced in (2). Now, the initial set of selected routes is $\mathcal{S} = \{R_1^\star\}$, and Bob removes route $R_1^\star$ from $\mathcal{R}$. Then, Bob repeats the following two steps until it selects a total of N_c routes.

Step 1–For each route r left in set $\mathcal{R}$, Bob forms a temporary set $\mathcal{T}_r = \mathcal{S} \cup \{r\}$ and computes the fairness of the number of hops in this set via Jain's formula, i.e.,

$$J(\mathcal{T}_r) = \left(\sum_{i\in\mathcal{T}_r} L_i\right)^2 \left(|\mathcal{T}_r| \sum_{i\in\mathcal{T}_r} L_i^2\right)^{-1}. \tag{4}$$

This yields $|\mathcal{R}|$ fairness values, which Bob groups in set $\mathcal{R}'$.

Step 2–Bob extracts a subset $\mathcal{M}' \in \mathcal{R}'$ including up to N_F routes $r \in \mathcal{R}$ that yield the least fairness. N_F is a predefined parameter known to Alice and Bob, and we assume it is also known to Eve. Now, Bob groups such fairness-equivalent routes in the set $\mathcal{M}''$, chooses the route most disjoint from those previously selected (using the metric (2)), and adds this route to the set $\mathcal{S}$. Then, if multiple routes still remain, Bob applies any deterministic rule to pick one route. We assume that both Alice and Eve know such rule.[4]

C. Choice of the Node ID

To conceal topology data from Eve, all intermediate nodes choose a temporary local ID. This ID changes between the upstream flooding of R-Pings to Bob and the downstream flooding of R-Reps to Alice, and for all subsequent flooding processes. This way, Eve's knowledge only comes from the R-Pings and R-Reps relayed by her neighbors. We note that even Alice and Bob have no prior information about such local IDs, nor do they require this information for key generation. Hence, the ID information is not considered concealed, and, consequentially, there is no risk of information leakage. To limit both the ID length and the probability that two nodes choose the same temporary IDs, we periodically generate disjoint pools of numbers, one pool for each UWAN node. With this solution, no two nodes ever pick the same ID, and the probability that a node picks the same ID twice in a row can be made arbitrarily small. Random node IDs are key to avoid leaking information to Eve about i) which and how many nodes are deployed, and thus ultimately ii) how far they can be located. The latter is particularly important, because it can

[4]For example, Bob may order all routes lexicographically in terms of the node IDs in each route, and pick the first route in this order. Alternatively, it can choose the route whose R-Ping arrived first.

translate into extra information on the maximum propagation delay expected, thus on the maximum size of the key obtained after each round. If this information were available, Eve could attempt to bound the maximum number of routes, thereby assisting in the network topology reconstruction through MDS. However, note that using temporary IDs is not essential for the key generation scheme itself, so long as Eve is not in a position to overhear each and every single transmission in the network. That is, the purpose of the random IDs is to augment the secrecy of our solution by making it harder for Eve to calculate the maximum propagation delay and to link between packets transmitted during R-Ping and during R-Rep for better propagation delay estimation in the hop. However, as a worst-case approach, we note that our theoretical analysis below does not account for this randomization.

D. Repeated Key Generations

As shown in Sections VI and VII, the number of secret bits may be limited for reasonable values of ρ (say, 0.1 s). To obtain a longer sequence, the key generation process is repeated K times, and we call each generation process *round*. Note that the sequences obtained over multiple rounds can be correlated if the network topology changes slowly across consecutive rounds, as discussed in more detail in the following.

E. Key Distillation

At the end of the K rounds, the *key distillation* step would typically follow. This process includes information reconciliation and privacy amplification. Information reconciliation exploits error-correcting codes to align the possibly different bit sequences obtained by Alice and Bob after the K rounds. We remark that reconciliation is best performed at the end of all rounds. Conversely, reconciling multiple times (e.g., once after each round) is suboptimal, because the employed error-correcting codes generally work better with longer sequences. After reconciliation, the privacy amplification phase extracts a sequence of $n < M$ bits truly secret to Eve (cf. [3, Ch. 4]). This typically involves hashing algorithms applied to the bit sequence. An in-depth analysis of the information reconciliation and privacy amplification phases is out of the scope of this work.

Note that our scheme does not depend on channel characteristics, but only assumes that the propagation delay from Alice to Bob is equal in both directions (down to the quantization step ρ). Also, we do not require the discovery process to reveal the entire network: in fact, depending on the location of Alice and Bob, different, possibly partial sets of links may be probed to compute propagation delays. However, our scheme assumes that all relay nodes between Alice and Bob can accurately measure their hardware and software processing delay.

V. Secrecy Metrics

We now present our attacker model and a suitable attack strategy for Eve to infer the secret key, under the proposed key agreement protocol. Then, we analyze the performance of the secret key agreement procedure under this attack.

A. Eve Model and Attack Strategy

Eve is a powerful overhearing device, that can receive and decode incoming packets. For example, this is the case when Alice and Bob communicate through a publicly known physical layer scheme, e.g., according to the JANUS standard [40]. We assume that Eve aims at reconstructing the secret key, but does not transmit any packet to avoid revealing her presence. Finally, the authenticity of data packets in the UWAN is fully ensured [16] and an impersonating attack is not considered. We note that Eve can overhear all communications within her range, including the requests and replies from Alice and Bob and the packets forwarded by the relay nodes. We also note that our model accommodates the possibility that Eve is more powerful than the UWAN nodes, e.g., being equipped with an array of microphones or having higher-quality equipment. We allow Eve to gain all the information available to Alice and Bob, to successfully decode all packets detected by Eve, to be time-synchronized with Alice and Bob, and to evaluate the time of arrival of incoming packets with no errors. However, Eve is not given information that is also concealed from Alice and Bob, e.g., the network's topology and nodes locations.

The objective of Eve is to observe the R-Ping/R-Rep exchange used for key generation, in order to infer the same key derived separately by Alice and Bob. For this purpose, Eve intercepts R-Pings and R-Reps by promiscuously listening to transmissions. This way, Eve gets exposed both to the route traveled by these packets, and to the accumulated processing delay stored therein, and can estimate the propagation delay accumulated by each packet. We assume that Eve knows all public information, such as the route selection method, N_c, ρ, N_r, T_c, and the algorithm of Fig. 1. We further make the generous assumption that Eve is sufficiently powerful to synchronize itself with both Alice and Bob, despite the fact that Eve is an eavesdropper. Thus, Eve can compute the ToFs of all R-Ping and R-Rep packets. Moreover, this relieves the need for Eve to know its own location. Yet, Eve is located at a random location, and overhears only a subset of the transmissions.

During the R-Ping forwarding phase, Eve logs all intercepted R-Ping packets and their reception times, and reconstructs routes by merging R-Pings with R-Reps. Unless only either Alice or Bob appear in the ID list of the R-Ping or R-Rep, Eve only merges packets with mutual intermediate nodes she is connected to, and observes different propagation times than Alice and Bob due to her location in the network and her distance from her neighbors. Eve can localize a relay via methods akin to matched field processing [41], if she knows both the bathymetry and the sound speed, and the bathymetry is sufficiently diverse around her. In any case, Eve can only detect locally overheard packets, and may thus fail to gather enough information if the UWAN topology is sufficiently sparse. Considering this, Eve may turn to graph localization methods, where holes due to partial knowledge of the network topology can be filled by setting upper and lower constraints on the link's delays. Approaches similar to multidimensional scaling (MDS) can be adopted, which reconstruct node coordinates from pairwise distance measurements [5]:

indeed, a relevant example is given by our recent scheme for UWANs [42], which allows Eve to reconstruct the network topology and improve her attack performance.

B. Generated Sequence Model

Let $\boldsymbol{X}(k) = [x_1(k), \ldots, x_{N_c}(k)]$ and $\boldsymbol{Y}(k) = [y_1(k), \ldots, y_{N_c}(k)]$ be the N_c net *quantized* route delays measured at round k by Alice and Bob, respectively. Let also $\boldsymbol{Z}(k) = [z_1(k), \ldots, z_{N_c}(k)]$ be the corresponding delays measured by Eve. We assume that the route delays are independent and identically distributed over the rounds: this is achieved when the rounds are separated by a time that yields enough variations in the position of the nodes.[5] From the sequences $\bar{\boldsymbol{X}} = [\boldsymbol{X}(1), \ldots, \boldsymbol{X}(K)]$ and $\bar{\boldsymbol{Y}} = [\boldsymbol{Y}(1), \ldots, \boldsymbol{Y}(K)]$ collected over K rounds, Alice and Bob will obtain the *secret key* $\mathcal{K}(K)$, as a result of the whole key agreement process, including advantage distillation and privacy amplification. This is the *source model* for secret-key generation [3, §4.3], where Alice, Bob, and Eve observe realizations of random, correlated bit sequences.

C. Secrecy Analysis

We now analyze the secrecy of the proposed method. Secrecy here refers to the fact that the (normalized) mutual information between the key and the sequences obtained by Eve $\bar{\boldsymbol{Z}} = [\boldsymbol{Z}(1), \ldots, \boldsymbol{Z}(K)]$ goes to zero as $K \to \infty$, i.e.,

$$\lim_{K\to\infty} \frac{1}{K} \mathbb{I}(\mathcal{K}(K); \bar{\boldsymbol{Z}}) = 0, \tag{5}$$

where $\mathbb{I}(\boldsymbol{a}; \boldsymbol{b})$ is the mutual information between the two random sequences $\boldsymbol{a}$ and $\boldsymbol{b}$. If $\mathcal{K}(K)$ includes n bits, and the k rounds took a total amount of time KT, the rate of the key (in bit/s) is $R = n/(KT)$. The *weak secret-key capacity* is the maximum rate R of the key that remains secret to Eve, i.e., the maximum rate for which (5) still holds.[6]

The following theorem bounds the secret-key capacity, considering quantized delay sequences [3, Theorem 4.1]:

Theorem 1: The weak secret-key capacity C_s of a source model $(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}, \bar{\boldsymbol{Z}})$ satisfies

$$\begin{aligned} &\mathbb{I}(\bar{\boldsymbol{X}}; \bar{\boldsymbol{Y}}) - \min\{\mathbb{I}(\bar{\boldsymbol{X}}; \bar{\boldsymbol{Z}}), \mathbb{I}(\bar{\boldsymbol{Y}}; \bar{\boldsymbol{Z}})\} \leq C_s \\ &\leq \min\{\mathbb{I}(\bar{\boldsymbol{X}}; \bar{\boldsymbol{Y}}), \mathbb{I}(\bar{\boldsymbol{X}}; \bar{\boldsymbol{Y}} | \bar{\boldsymbol{Z}})\} \,. \end{aligned} \tag{6}$$

Moreover, if we exploit the channel reciprocity in terms of the route's propagation delay (i.e., $\bar{\boldsymbol{X}} = \bar{\boldsymbol{Y}}$), and we assume no errors in the delay estimation, we can simplify the above expression as

$$\mathbb{H}(\bar{\boldsymbol{X}}) - \mathbb{I}(\bar{\boldsymbol{X}}; \bar{\boldsymbol{Z}}) \leq C_s \leq \mathbb{H}(\bar{\boldsymbol{X}} | \bar{\boldsymbol{Z}}). \tag{7}$$

Then, from the definition of mutual information, we have

$$\mathbb{I}(\bar{\boldsymbol{X}}; \bar{\boldsymbol{Z}}) = \mathbb{H}(\bar{\boldsymbol{X}}) - \mathbb{H}(\bar{\boldsymbol{X}} | \bar{\boldsymbol{Z}}) \,. \tag{8}$$

Hence, the bounds in (7) coincide, yielding the identity

$$C_s = \mathbb{H}(\bar{\boldsymbol{X}} | \bar{\boldsymbol{Z}}) \,. \tag{9}$$

[5]The number of secret bits obtained through the K rounds would decrease in case the route delays are correlated.

[6]In addition to (5), the reliability and weak uniformity constraints must also hold (see [3, Definition 4.3] for details): these are satisfied in our context.

To write the entropy of the quantized delay sequences, we recall that they are obtained by the quantization of the delays, which are continuous random variables. Let $\bar{\boldsymbol{X}}'$, $\bar{\boldsymbol{Y}}'$, and $\bar{\boldsymbol{Z}}'$ be the continuous random vectors corresponding to $\bar{\boldsymbol{X}}$, $\bar{\boldsymbol{Y}}$, and $\bar{\boldsymbol{Z}}$, respectively. For vectors of continuous random variables, we recall the definition of *differential entropy*

$$h(\bar{\boldsymbol{X}}') = \int p_{\bar{\boldsymbol{X}}'}(\boldsymbol{x}) \log_2 p_{\bar{\boldsymbol{X}}'}(\boldsymbol{x}) d\boldsymbol{x}, \tag{10}$$

where $p_{\bar{\boldsymbol{X}}'}(\boldsymbol{x})$ is the *probability density function* (PDF) of $\bar{\boldsymbol{X}}'$. Then, we consider a uniform quantizer for each entry of $\bar{\boldsymbol{X}}'$, $\bar{\boldsymbol{Y}}'$, and $\bar{\boldsymbol{Z}}'$, with step ρ. Hence, if ρ is small enough, we can approximate the entropies as (see [43, Theorem 8.3.1])

$$\begin{aligned} \mathbb{H}(\bar{\boldsymbol{X}}) &\approx h(\bar{\boldsymbol{X}}') - KN_c \log_2 \rho, \\ \mathbb{H}(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Z}}) &\approx h(\bar{\boldsymbol{X}}', \bar{\boldsymbol{Z}}') - 2KN_c \log_2 \rho. \end{aligned} \tag{11}$$

Hence, from (9) we obtain

$$\begin{aligned} C_s(K) &= \mathbb{H}(\bar{\boldsymbol{X}} | \bar{\boldsymbol{Z}}) \approx \mathbb{H}(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Z}}) - \mathbb{H}(\bar{\boldsymbol{Z}}) \\ &= h(\bar{\boldsymbol{X}}', \bar{\boldsymbol{Z}}') - h(\bar{\boldsymbol{Z}}') - KN_c \log_2 \rho, \end{aligned} \tag{12}$$

which means that, as long as the reciprocity of the channel is verified, and no errors occur in the delay estimation, Alice and Bob can increase the number of secret bits by simply picking a finer uniform quantizer.

Remark: From (11), we note that another option to increase the length of the secret keys is to increase the number of quantization bits, i.e., via a small value of ρ.

Remark: When the time elapsed between K consecutive sequence-generation rounds is small, elements $\boldsymbol{X}(k), k = 1, \ldots, K$, may be correlated. This dependency is taken into consideration in (10) by the joint distribution $p_{\bar{\boldsymbol{X}}'}(\boldsymbol{x})$. Thus, the number of secret bits is upper bounded by $KC_{\mathrm{s}}(1)$.

Remark: From (9), the secret key capacity depends on the relation between the observations by Alice and Eve. In particular, the more accurate Eve's reconstruction of the topology is, the lower the conditional entropies will be. If Eve can overhear all packets, $\bar{\boldsymbol{X}} = \bar{\boldsymbol{Z}}$, the secret key capacity drops to zero, as Alice and Bob would not have any advantage over Eve. More generally, when Eve is equipped with sufficiently sophisticated devices (e.g., very directive microphone arrays), her reception capabilities improve, possibly covering all transmitted messages and making it impossible for Alice and Bob to obtain secret keys. Eve's position also affects her reception capabilities and hence the secret key capacity: in fact, when Eve is very close to Alice or Bob, $\bar{\boldsymbol{X}}$ and $\bar{\boldsymbol{Z}}$ become very similar, reducing the secret key capacity.

Remark: Lastly, in this paper we considered a passive Eve that only overhears packets. A more challenging scenario occurs if Eve can inject packets, e.g., because no authentication mechanism is in place. In this case, Eve can determine part of the agreed sequence, as she may flood her own packets into the network. Still, some routes will not pass through Eve during the flooding process, thus an alteration of the entire sequence is not possible. A detailed analysis of the performance of such advanced attacks is postponed to future work.

We note that the operation of our scheme requires the setting of a few system parameters. Some guidelines on how to set these parameters follow. The globally known maximum

propagation time from Alice to Bob, T_c, is a function of the number of nodes in the network and the maximum propagation delay plus hardware and re-transmission delay in a link. While we do not assume knowledge of the number of nodes in the network, T_c can be set by dividing the explored area into area units the length of the range specification for the used acoustic modem and considering the maximum route between these units. Similarly, the globally known number of chosen routes, N_c, can be a function of the number of such area units, N_a, and be set to a maximum of $\log_2 N_a$ routes as a trade-off between the choice of missing a route and diversity of routes. The quantization parameter, ρ, should be set as the sum of the assumed time synchronization offset between Alice and Bob, the expected error in the time-of-arrival measurement, which in turn can be assumed to be the root-mean-square of the channel's multipath delays, and the expected change of the propagation delay between the Alice-Bob link and the return Bob-Alice link, which can be set by the expected motion of the nodes. The number of key generation attempts, K, is a function of the length of the requested key.

VI. Numerical Simulations

In this section, we present results from numerical simulations. Since the focus of our work is the generation and exchange of secret key, we consider only the key exchange part of the communication session and avoid data encryption and data transmission. In our simulations, we neglect any losses of R-Ping and R-Rep packets. However, we demonstrate in the lake experiment that, using an underlying error control mechanism, such failures have no effect on performance. We use $T_c = 30\,\mathrm{s}$ as the waiting time for the collection of R-Pings and R-Reps, $K = 1$ round, $N_c = 1$ or $N_c = 3$, and $N_r = \lceil \log_2(30/\rho) \rceil$, where the delay quantizer step ρ is changed to explore the algorithm's sensitivity. Note that, with the above numbers, the propagation delay for each route can be represented as a sequence of 9 bits for $\rho = 0.1$ s, 8 bits for $\rho = 0.2$ s, and 7 bits for $\rho = 0.4$ s. The number of nodes, N, is pre-set but is considered unknown for Alice, Bob, and Eve. For Eve, we implemented the attack strategy described in Section V-A.

We consider two setups for our simulations: 1) a random static network topology, where nodes are randomly placed at stationary locations, link attenuation is ignored, and ToF measurements are assumed accurate, e.g., by using the process in [44] (Section VI-A); and 2) a communication network simulated by using the well-known Bellhop acoustic propagation model [45], including mobility (Section VI-B). Setup 1 enables a statistical test for the number of potential secret bits obtained from our chosen sources of randomness, whereas the latter explores the key agreement rate in a more realistic environment, including ToF measurement errors. In both simulations, all nodes are deployed uniformly at random over an area of 3 km $\times$ 3 km, with Alice, Bob, and Eve chosen randomly among these nodes. The communication range of the nodes is set to 1 km. Moreover, R-Pings from Alice and R-Reps from Bob are propagated through the network by assuming a sound speed of 1500 m/s, and a hardware/software delay randomly distributed between 0 and 1 s for each node.

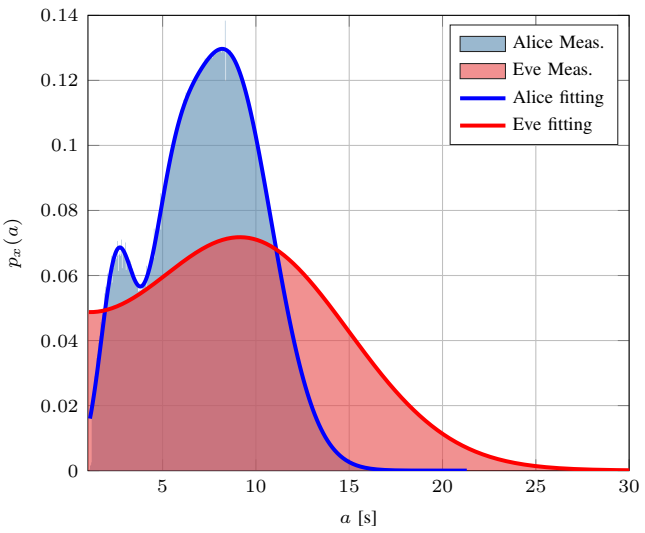


Fig. 2. Empirical distribution of Alice (blue) and Eve (red) measurements and their estimated PDFS (thick lines).

We assume that Eve has the same communication equipment of the other UWAN network nodes, and we leave the evaluation of the case where Eve has more sophisticated equipment for future studies. Moreover, we do not assume that Eve can choose her position to maximize the number of overheard packets, but we select her position at random, as for the UWAN nodes. Finally, we remark that we mostly focus on evaluating the performance for a single round in what follows. Indeed, one round of our scheme constitutes a basic building block, and Alice and Bob can construct longer keys by coalescing the bits extracted from multiple rounds, taking care of detecting and mitigating the correlation among the propagation delays over subsequent rounds, as discussed in Section V-B.

A. Random Network Topology

To analyze the performance of random network topologies, we deploy artificial blocks of length equal to 100 m uniformly at random in the network area. These blocks separate the network nodes such that communications between any two nodes are possible only if their line-of-sight link is unblocked and their distance is below 1 km. The result is a sparse network topology of $N = 6$ nodes. We collected a Monte-Carlo set of 17 million realizations of the above scenario. We then exploit the resulting large dataset to draw statistically significant conclusions about the information-theoretic security properties of our proposed secret key generation scheme.

Measurements model for Alice, Bob and Eve: Fig. 2 shows the distribution of the net route delay measurements $x_1(k)$ at Alice, in light red, and Eve, in blue. Alice's PDF was modeled as a clipped mixture of three Gaussian distributions, i.e.,

$$p_{x_1(k)}(x) = \sum_{n=1}^{3} A_n e^{-\frac{(x-\mu_n)^2}{\sigma_n^2}}, \quad x \in [x_{\min}, x_{\max}], \qquad (13)$$

and $p_{x_1(k)}(x) = 0$ if $x \notin [x_{\min}, x_{\max}]$. Note that the model matches the experimental distribution well by using only three

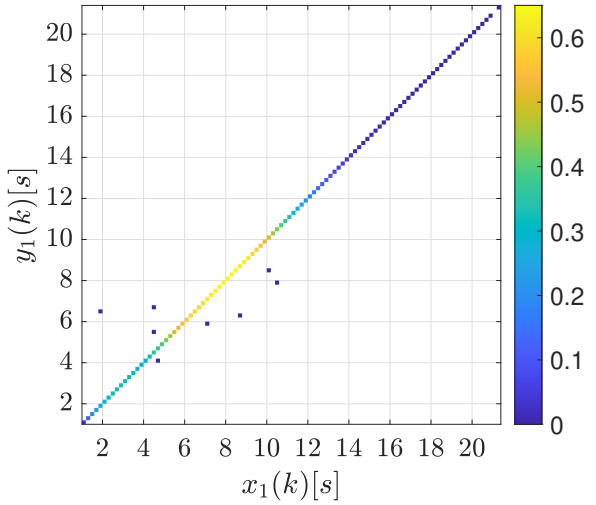


Fig. 3. Joint distribution of Alice and Bob measurements (top view).

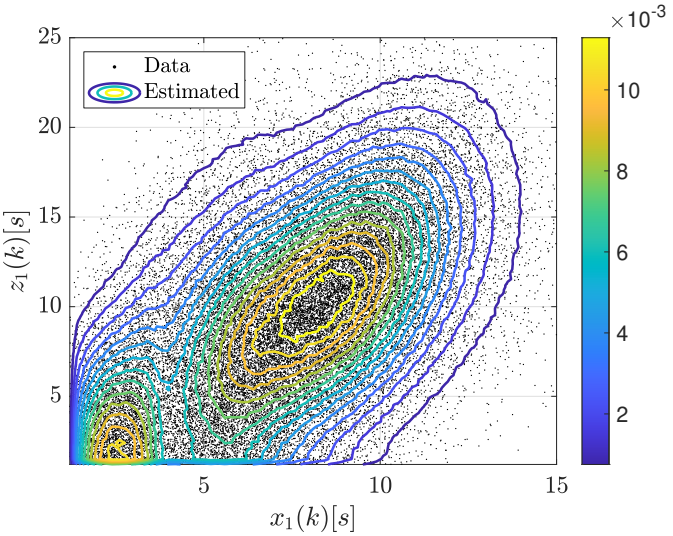


Fig. 4. Observed (black dots) and estimated (solid lines) joint PDF of Alice's and Eve's measurements.

Gaussian components. Similar results are obtained also for the delays of the other two routes ($x_2(k)$ and $x_3(k)$). Eve distribution was instead approximated as

$$p_{z_1(k)}(z) = \sum_{n=1}^{2} A'_n e^{-\frac{(z-\mu'_n)^2}{\sigma'^2_n}}, \quad z \in [z_{\min}, z_{\max}], \qquad (14)$$

and $p_{z_1(k)}(z) = 0$ if $z \notin [z_{\min}, z_{\max}]$. Again the model matches the experimental distribution.

Joint Alice and Bob Measurement Model: Fig. 3 shows the joint distribution of Alice's and Bob's measurements, $x_1(k)$ and $y_1(k)$. We clearly see that the assumption of the channel reciprocity is verified in these simulations, hence in practice we may consider skipping the information reconciliation step; in more detail, for a dataset composed of ≈ 17 million measurements we have that $P(x_i(k) \neq y_i(k)) \simeq 5{\times}10^{-7}$.

Joint Alice and Eve Measurements Model: By way of contrast, Fig. 4 shows the joint PDF $p_{x_1(k),z_1(k)}(x,z)$ of delay measurements by Alice and Eve. Collected measurements appear as black dots; the joint PDF was computed by (dense) linear interpolation. When compared to the Alice-Bob measurements of Fig. 3, we notice a much weaker relationship between Alice's and Eve's measurements, since the joint distribution $p_{x_1(k),z_1(k)}(x,z)$ is much more sparse. Hence, it is hard for Eve to exploit her own measurements to infer those of Alice.

The first two rows of Table I show the results for a single round ($K = 1$) and $N_c = 1$. Results are shown considering one of the 3 obtained routes, where index $i = \{1, 2, 3\}$ indicates the selected route. We neglect the term due to quantization, $-N_c \log_2 \rho$, to highlight the contribution of the channel rather than the quantizer to the secret key rate. Notice that we did not compute entropy and capacity for $N_c = 3$ due to the prohibitive computational cost of such an operation.

The last two rows of Table I show instead the performance results, again for $K = 1$, but for both $N_c = 1$ and $N_c = 3$. To compute the capacity we use (12), which requires the estimation of a 6-dimensional distribution, since $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$ are

TABLE I

$C_s(1)$ COMPUTED USING THE DIFFERENTIAL ENTROPY $h(x_i(k))$ AND ENTROPY $\mathbb{H}(\bar{\boldsymbol{X}})$, FOR $K = 1$ AND $N_c = \{1, 3\}$ AND THREE ROUTES (INDEX i). $\mathbb{H}(X)$ AND THE ASSOCIATED CAPACITY $C_s(1)$ ARE COMPUTED BY FIXING THE QUANTIZATION STEP TO $\rho = 0.4$ S.

	$N_c = 1$			$N_c = 3$
	$i = 1$	$i = 2$	$i = 3$	
$h(x_i(k))$ [bit]	3.534	3.556	3.546	—
$C_s(1)$ [bit]	3.245	3.244	3.207	
$\mathbb{H}(\bar{\boldsymbol{X}})$ [bit]	4.519	4.139	3.986	12.05
$C_s(1)$ [bit]	4.252	3.807	3.575	9.865

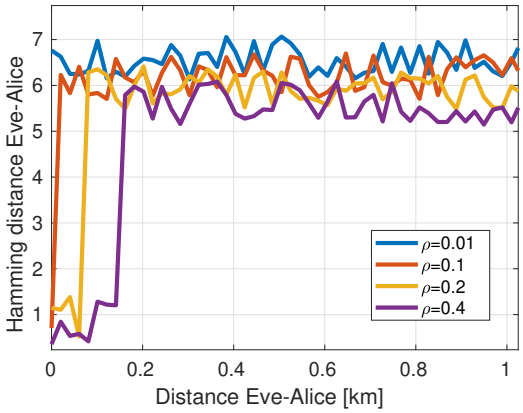


Fig. 5. Hamming distance between Eve and Alice keys as a function of the Eve-Alice range.

vectors of length 3. To maintain a reasonable computational complexity, in this case, we set $\rho = 0.4$ s. Even if the values differ from the continuous case by a value which is slightly less than $-\log_2 \rho$, we still notice that the entropy obtained in the latter case is close to the sum of the entropies obtained from every single route: thus, we conclude that the delays are close to being statistically independent.

Fig. 5 shows the Hamming distance between the keys

generated by Eve and Alice as a function of their distance. If located closer to Alice, Eve can intercept a larger number of R-Pings and R-Reps, and thus reconstruct the network topology more accurately. Results would be entirely equivalent if Eve were located close to Bob. We observe that at distances farther than $\rho \cdot c$ where c in the sound speed in water the Hamming distance is large, and no correlation is observed between the Hamming distance and the Eve-Alice range. This is due to the hash function used to generate the key from the time-of-arrival measurements. On closer distances, the Hamming distance is much smaller. This indicates that, when Eve is close to Alice with respect to ρ, she can estimate the same route propagation delays Alice's. We remark that similar results would apply if Eve has a better receiver than Alice's or Bob's, e.g., including a larger array enabling a more directive reception.

B. Bellhop Results

In addition to the Monte-Carlo simulations, we also collected results from simulations in fully mobile network scenarios, as typically found in mobile networks of underwater drifters deployed to sample or monitor some physical phenomenon [46]–[48], as well as in coordinated missions involving multiple autonomous underwater vehicles (AUVs) [49]–[51]. For this evaluation, we resort to the Bellhop ray tracing software [45] to simulate the acoustic propagation between pairs of network nodes. We deployed 12 nodes along the four sides of a square, three nodes per side. The side length is 3 km. We chose environmental parameters (bathymetry, sediments, and sound speed profile) from the bay area of San Diego, CA, USA. The south-western side of the square has longitude and latitude coordinates $(-117.3661, 32.8628)$ degrees. After the deployment, the nodes move at a nominal speed of up to 1 m/s towards the opposite side of the square. Once they reach it, they move back towards their initial side and keep repeating the same pattern for the whole duration of the simulation. Besides the nominal speed component, we add a random Gauss-Markov component that accounts for local perturbations in the North-South and East-West directions. These perturbations include a current having an average speed of 0.5 m/s that makes all nodes drift North. The above settings and the initial square arrangement ensure that the network topology varies to include both sparse deployments (e.g., when the nodes are closer to the sides of the square) and denser ones (e.g., when the nodes move towards the center). We remark that the random Gauss-Markov components simulate drifts, thus, the shape of the topology changes over time. We assume that the mobile nodes can keep their depth constant to 50 m through proper depth gauges and actuation. This is typical of many sufficiently advanced AUVs. Each node has a constant communication range of about 1700 m.

For each Bellhop run, we model the links between the different nodes by implementing the JANUS underwater acoustic communication protocol [40] for the R-Ping and R-Rep messages, and by convolving the modulated signals with the current channel realization. White noise is added to the outcome such that the signal-to-noise ratio is 10 dB. Then, running the JANUS receiver, we determine a link to be active if the bit error rate is below 10^{-3}. In turn, the link delay is set by the arrival time of the first channel tap, as received from the Bellhop realization. As Bellhop accurately replicates the multipath structure of the channel, it becomes possible that different multipath patterns affect R-Ping and R-Rep transmissions, due to the movement and drift of the nodes. These changes may lead Alice and Bob to measure different end-to-end propagation delays, but can be compensated for through a higher quatization step ρ, if needed. We note that, in most cases, such an addition to ρ is negligible compared to the compensation required to manage time synchronization between Alice and Bob.

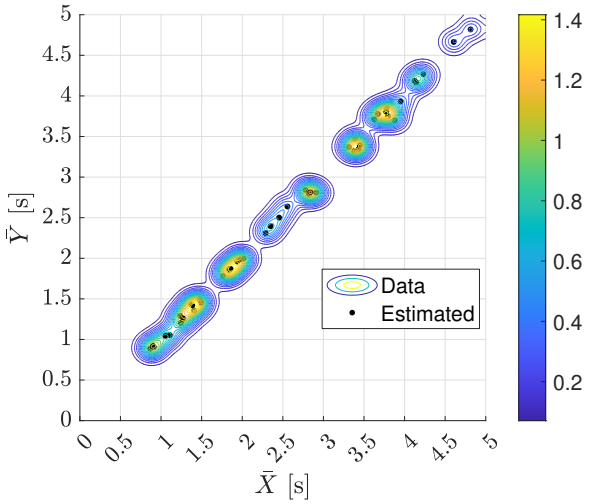


Fig. 6. Joint PDF of route delay measurements at Alice and Bob: Bellhop results (black dots) and PDF estimated via KDE (contour plot) for $v = 0.1$ m/s and $N_c = 1$.

Bellhop simulations are inherently more complex and lengthy than those described in Section VI-A, resulting in a dataset of less than one thousand secret-key agreement rounds. Therefore, we cannot directly estimate the PDFs as we did for the random topology simulations; instead, we resort to the well-known Gaussian *kernel density estimation* (KDE).

For this scenario we first consider $N_c = 1$ and $K = 1$, thus $\bar{\boldsymbol{X}}$, $\bar{\boldsymbol{Y}}$, and $\bar{\boldsymbol{Z}}$ become the scalar $\bar{X}$, $\bar{Y}$, and $\bar{Z}$. We now summarize the results of our statistical analysis on Bellhop simulation data. The results for the estimation of the joint PDF of delay measurements at Alice and Bob, $p_{\bar{X},\bar{Y}}(x, y)$, and of the joint distribution of delay measurements at Alice and Eve, $p_{\bar{X},\bar{Z}}(x, z)$, are reported in Figs. 6 and 7. We observe that all the conclusions drawn from the random network topology simulations still hold. From Fig. 6 we notice that, except for a few outliers, there is a very strong correlation between Alice and Bob's delay measurements, as all pairs of measurements are close to the $\bar{X} = \bar{Y}$ line; from Fig. 7, instead, we see that the KDE outputs a much smoother PDF, hinting the fact that the measurements collected by Alice and Eve are only weakly correlated. The cases where the MDS-based algorithm fails to give an estimate are reported as $\bar{Z} = 0$ and ignored from the PDF estimation.

Considering a single round as in (12) and $N_c = 1$, i.e., one

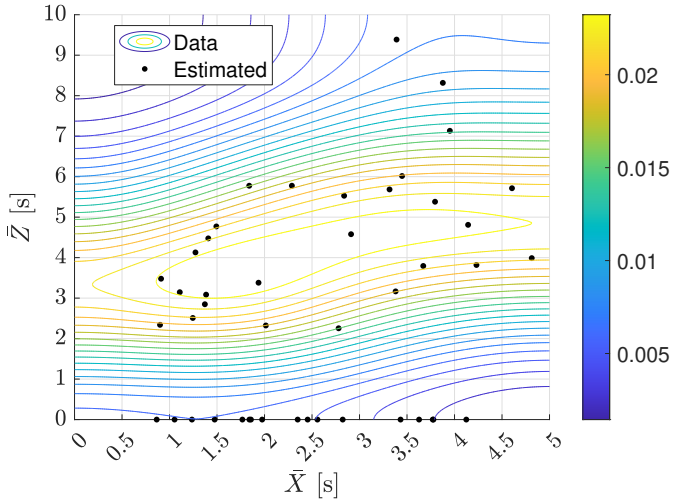


Fig. 7. Joint PDF of route delay measurements at Alice and Eve: Bellhop results (black dots) and PDF estimated via KDE (contour plot) for speed $v = 0.1$ m/s and $N_c = 1$.

route at a time but neglecting the factor due to quantization, $-\log_2 \rho$ we obtain a secrecy capacity $C_\mathrm{s} = 3.108, 3.099$ and 3.150 bit, respectively for routes $i = 1, 2$ and 3 with node speed of 0.1 m/s. Note that we can obtain a longer key either by picking a smaller quantization step, or by performing more rounds. The results are compatible with Table I.

We now consider a scenario with $K = 3$ and $N_c = 1$. Based on the simulated drift of the network nodes, we explore the effect of possible correlations between the delay observations $\boldsymbol{X}(k)$ in each key-generation round k. Table II summarizes the number of secret bits, $C_\mathrm{s}(K)$, according to (12), with several values of ρ. We compare these numbers to the upper bound $KN_cC_\mathrm{s}(1)$, while considering for $C_\mathrm{s}(1)$ the average value over the 3 routes, second row of Table I. The results are obtained by concatenating the observation delays, defined as the sum of the delays along all chosen routes, measured during 3 consecutive key-generation rounds, each lasting 30 s. In this context, it is important to observe the average difference between the accumulated delays between consecutive key-generation rounds of Alice,

$$\Delta_{\bar{\boldsymbol{X}}}(K) = \sum_{k=2}^{K} \left(\left| \sum_{i=1}^{N_c} x_i(k-1) - \sum_{i=1}^{N_c} x_i(k) \right| \right) , \qquad (15)$$

and of Eve,

$$\Delta_{\bar{\boldsymbol{Z}}}(K) = \sum_{k=2}^{K} \left(\left| \sum_{i=1}^{N_c} z_i(k-1) - \sum_{i=1}^{N_c} z_i(k) \right| \right) . \qquad (16)$$

Table II reports results for different values of the nodes' speed $S \in \{0.3, 0.6, 0.9\}$ m/s. We observe that, due to the diversity increase, $\Delta_{\bar{\boldsymbol{X}}}(K)$ increases with S, and as a result, so does $C_\mathrm{s}(K)$. In particular, the obtained $C_\mathrm{s}(K)$ is close to the bound $KC_\mathrm{s}(1)$, especially for higher values of the node speed. Still, this comes at the risk of key mismatches or at the cost of increasing the quantization step ρ. We also observe that $\Delta_{\bar{\boldsymbol{Z}}}(K)$ increases with S. This is because an increase in $\Delta_{\bar{\boldsymbol{X}}}(K)$ makes it harder for Eve to infer the correct observation delays. The above quantities make it possible to compute a lower bound[7] to the number of rounds required to achieve $b^\star$ secret bits using our scheme as

$$\widehat{K}(b^\star) = \left\lceil \frac{b^\star}{N_c\big(C_s(1) - \log_2 \rho\big)} \right\rceil . \qquad (17)$$

Assuming that the quantization bin size is $\rho = 0.3\,\mathrm{s}$, then $-\log_2 \rho \approx 1.737$. For $N_c = 3$ and a node speed of 0.9 m/s, we need $\widehat{K}(256) = 18$ rounds to accrue 256 secret bits. Assume $T_c = 15\,\mathrm{s}$ is the upper bound on the time it takes to propagate packets from Alice to Bob and vice versa in all the N_c chosen routes. Repeating this process for $K = 18$ times with a $\mu_s = 30\,\mathrm{s}$ delay between each key generation session, then 256 secret bits (from which a much largest key is formed) are obtained after $18\,\mathrm{min}$. This delay is perfectly acceptable in underwater networks, that may operate unattended for several weeks [52]–[55].

In Figs. 8a and 8b we analyze the impact of the number of routes, R_k, and the number of hops, L_k, respectively, on the Hamming distance between the quantized ($\rho = 0.3$ s) route delay vectors of Alice and Bob, denoted as $d_H(\boldsymbol{X}(k), \boldsymbol{Y}(k))$, and of Alice and Eve, $d_H(\boldsymbol{X}(k), \boldsymbol{Z}(k))$. Here, $N_c = 3$. We observe that the agreement between Alice and Bob is hardly affected by the network structure. This is because, although the nodes are slowly drifting in each simulation run, the network topology changes during the delay measurement procedure are negligible. As a result, Alice and Bob are expected to choose similar routes (3 routes in our simulations) out of all routes available, and hence their performance in terms of sequence generation is not affected. Depending on the specific network setting and the surrounding environments, faster speeds would increase the likelihood that Alice and Bob measure different route propagation times for the same routes, or that routes change between Alice's and Bob's measurements. Consequently, the resolution parameter ρ should be aligned with the assumed maximum speed of the nodes.

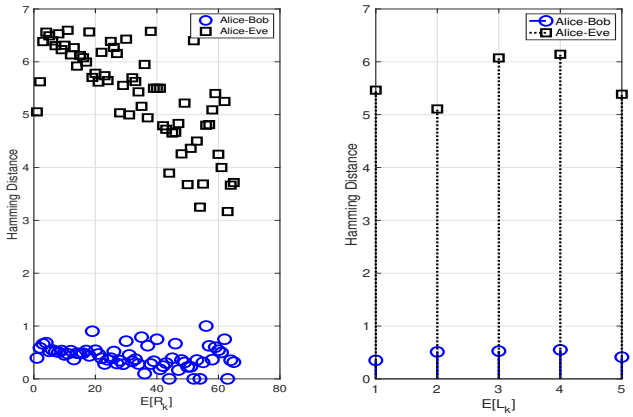
However, the results show that the capability of Eve to extract the secret key is affected by the network's structure. In particular, the Hamming distance between the Alice and Eve bit sequences decreases with the number of routes and slightly increases with the number of hops. This is because the availability of more routes is correlated with denser networks, characterized by a higher degree of connectivity among the nodes. Then, Eve has more chances to overhear more communication sessions and thus extract more information to feed into the MDS-based algorithm she uses to infer the key. However, as the number of hops increases, the network is sparser and the nodes' connectivity decreases. Then, it is harder for Eve to discover the true topology of the network and its capability to infer the key decreases.

In the next section, we complement the above conclusions through the results of a dedicated lake experiment.

[7]If Alice and Bob unrecoverably fail to agree on a key (e.g., because some R-Ping or R-Rep packets get lost, or because propagation delays change significantly between the R-Ping and R-Rep forwarding processes, the secret bit keys obtained in the current round are discarded.

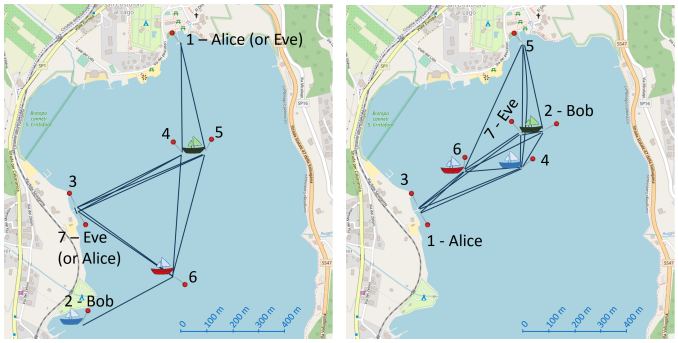
TABLE II
$C_s(K)$ FOR $N_c = 1$, AND $K = 3$, FOR $\rho = 0.4, 0.3, 0.2, 0.1, 0.06, 0.04$ AND 0.01 s. THE NETWORK INCLUDES 6 NODES.

Speed	$C_s(K)$ [bit]							$KN_cC_s(1)$ [bit]							$\Delta_{\bar{X}}(K)$	$\Delta_{\bar{Z}}(K)$
[m/s]	ρ =0.4 s	0.3 s	0.2 s	0.1 s	0.06 s	0.04 s	0.01 s	0.4 s	0.3 s	0.2 s	0.1 s	0.06 s	0.04 s	0.01 s	[s]	[s]
0.3	8.52	8.94	9.52	10.52	11.26	11.84	13.84	13.67	14.91	16.67	19.67	21.88	23.63	29.63	3.4	47.5
0.6	9.12	9.54	10.12	11.12	11.86	12.44	14.44	13.67	14.91	16.67	19.67	21.88	23.63	29.63	5.5	61.2
0.9	9.82	10.24	10.82	11.82	12.57	13.14	15.14	13.67	14.91	16.67	19.67	21.88	23.63	29.63	8.6	98.4



(a) vs. $E(R_k)$, $N_c = 3$. (b) vs. $E(L_k)$, $N_c = 3$.

Fig. 8. Hamming distance between delay vectors obtained by Alice, Bob and Eve, $d_H(\mathbf{X}(k), \mathbf{Y}(k))$.



(a) Topologies 1 (and 2) (b) Topology 3

Fig. 9. Topologies tested during the lake experiment.

VII. LAKE EXPERIMENT

A. Experiment Setup

To test our scheme in a realistic setting, we organized an experiment on the Caldonazzo lake, in northern Italy (46.033115°N, 11.236756°E), on June 23, 2021. The lake provides a calm acoustic environment, with a typically flat surface. and a variable depth of up to 50 m. The experiment ran under mostly sunny weather with moderate wind. Based on ranging trials, the estimated sound speed in water was 1495 m/s. We used seven EvoLogics S2CR modems as acoustic transceivers, working in the 18–34 kHz band [56]. These modems can transmit packets of up to 64 bytes at 1 kbit/s, using a $7/8$-rate Reed-Solomon code. We developed custom Python modules to drive the modems and implement the R-Ping/R-Rep exchange. We deployed the modems from three boats to a depth of 5 m, and from two piers to a depth of 2 m, and tested three different network topologies (see Fig. 9). We tuned the UWAN connectivity both by leveraging natural obstacles (e.g., the small promontory towards the south-west) and by tuning the transmit power of each device.

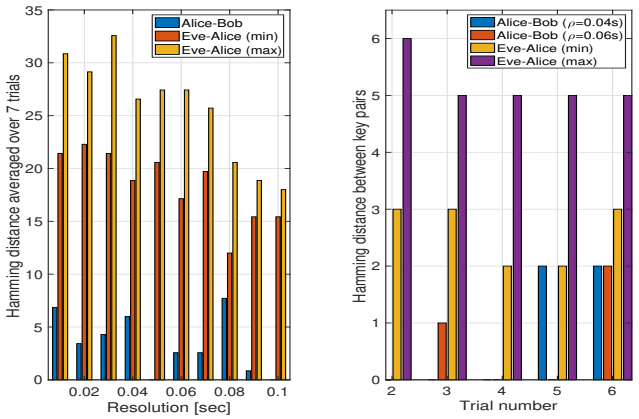
In Topology 1, Alice resides on the northern pier, and Bob is behind the southwest promontory. Alice and Bob can communicate only via multihop routes, e.g., through node 6. Eve is located on a pier to the west. We observed routes of up to 5 hops, providing a spread of about 500 ms between the minimum (900 ms) and maximum (1404 ms) route propagation delays. Topology 2 is the same as Topology 1, except that we exchanged the roles of Alice and Eve. In this case, the node on the northern pier also acts as a relay.

Topology 3 reunites all ship-based nodes in the middle between the west and the north pier so that all boat-deployed nodes are linked. Through software-based link blacklisting, we avoided Alice and Bob having a direct connection.

We performed 7 experiment rounds: one for Topology 1, two for Topology 2, and four for Topology 3. In the latter two cases, we observed changes due to the drift of the nodes. During each round, we ran the route discovery process sequentially, first from Alice (R-Ping) and then from Bob (R-Rep). For each route, we measure and accumulate the total retransmission delay not due to acoustic propagation as follows. We set up a cross-layer backoff-based collision avoidance scheme, whereby each relay waits for a random

TABLE III
AVERAGE NUMBER OF RETRANSMISSIONS BEFORE CORRECT RECEPTION OBSERVED OVER EACH LINK IN THE LAKE EXPERIMENT

	Topology 1							**Topology 2**							**Topology 3**						
↓TX / RX→	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7
1	—	—	—	0.00	0.00	—	—	—	—	0.95	0.17	0.31	—	0.00	—	—	—	0.00	—	0.00	1.21
2	—	—	—	—	—	0.00	—	—	—	—	—	—	0.00	—	—	—	—	0.00	0.56	0.20	0.00
3	—	—	—	0.83	0.71	1.21	1.07	0.92	—	—	0.71	0.52	—	0.03	—	—	—	0.50	—	0.63	0.86
4	0.91	—	0.59	—	—	0.48	0.68	0.43	2.00	0.76	—	0.59	0.81	0.94	0.42	0.36	0.54	—	—	0.29	0.39
5	0.88	—	0.69	—	—	1.32	0.80	0.63	—	0.40	0.43	—	0.82	0.56	—	—	—	—	—	—	—
6	—	0.57	0.70	0.26	0.39	—	0.68	—	0.71	—	0.70	0.92	—	0.50	0.58	0.21	0.41	0.28	—	—	0.42



(a) Results shown against ρ by rotating Eve across nodes $3, \ldots, 7$.

(b) Results of the seven field trials. $\rho = 0.04$ s and $\rho = 0.06$ s.

Fig. 10. Average Hamming distance between $\boldsymbol{X}(k)$ and $\boldsymbol{Y}(k)$ (Alice-Bob) and minimum and maximum distance between $\boldsymbol{X}(k)$ and $\boldsymbol{Z}(k)$ (Eve-Alice).

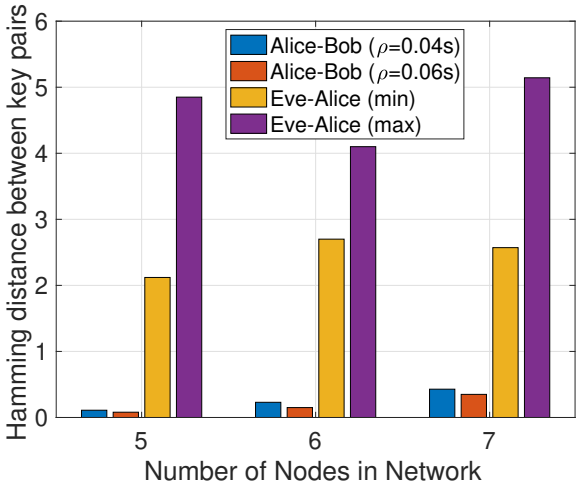


Fig. 11. Hamming distance averaged over the 7 trials as a function of the number of nodes in the network.

amount of time between 4 and 12 s before retransmitting a received R-Ping or R-Rep. The minimum delay of 4 s allows ample time for the software to operate. After drawing the backoff time at random, we read the reception time from the modems' "extended notifications," and send a so-called "synchronous instant message" at the exact expiry epoch of the backoff timer. This maximizes the accuracy of retransmission delay measurements. We compensate for losses (e.g., due to collisions or low-quality acoustic links), by repeating each R-Ping or R-Rep transmission three times, each preceded by a different random backoff. These settings enable recovering from transmission errors due to temporarily lower link quality, and at the same time prevent excessive delays due to prolonged transmission attempts on persistently low-quality links.

Table III summarizes the link-level performance throughout all experiments by reporting the average number of retransmissions required to communicate over each link. As expected, we observe a few links that show high reliability (0 retransmissions), a majority of links that occasionally require one retransmission before a packet is correctly received, and some that typically require more than one retransmission. In no case do the nodes ever need to retransmit more than twice, showing that setting the number of maximum retransmissions to 3 suffices in this environment.

Finally, all modems are connected to global positioning-system-(GPS)-synchronized controllers (either a laptop or a Raspberry-Pi3). This enables global time reckoning and allows us to process the experiment results by assuming that Eve can be co-located with any of the relay nodes.

B. Experimental Results

In all experiments, we set a maximum value of 30 s for the total propagation delay that R-Pings and R-Reps can incur while propagating in the network (note that this number does not include the backoff time at each relay, or the cumulated backoffs after multiple retransmissions). Moreover, we set $N_c = 3$, and therefore $N_r = \lceil \log_2(30/\rho) \rceil$, where ρ is changed to explore the algorithm's sensitivity. We start by exploring the sensitivity of the key agreement of Alice, Bob, and Eve to the resolution parameter, ρ. Fig. 10a shows the Hamming distance between the quantized delay vectors at Alice and Bob $d_H(\mathbf{X}(k), \mathbf{Y}(k))$, and those obtained by Alice and Eve $d_H(\mathbf{X}(k), \mathbf{Z}(k))$, as a function of ρ, summed over the seven rounds. Therefore, values smaller than 7 indicate a perfect key agreement in multiple rounds. For Eve, we also show results for the minimum and maximum key mismatches obtained by changing the identity of Eve between nodes 3-7. As expected, we observe that the number of quantized route delay bit mismatches between Alice and Bob reduces in principle with ρ. In fact, larger values of ρ make it possible to accommodate larger delays within the same quantization bins and hence help compensate for small but inevitable mismatches in the route delays. However, the coarser quantization comes at the expense of a shorter key.

Another way to observe this tradeoff is through Eve's quantized delay bit-sequence mismatches, which also decrease with ρ because of the shorter key length. Considering the results for Eve, we conclude that the proposed security algorithm is not significantly affected by the location of Eve across the network, due to the high number of delay sequence mismatches.

Next, we explore how the observed delay vectors change across the different rounds. Considering $\rho = 0.04$ s, Table IV shows the Hamming distance between the delay vectors at Alice and Eve $d_H(\mathbf{X}(k), \mathbf{Z}(k))$ across the seven trials. Hence, any change in the delay sequences between trials tested under the same topology (as marked by different colors in Table IV) is due to the drift of the nodes. We observe that, for Alice, the delay vectors change considerably by at least 4 bits across different topologies. Also, significant differences emerge between round 1 of Topology 3 and rounds 2 to 4, as well as between round 4 and all the previous rounds of Topology 3. However, this change is not observed across the two rounds of Topology 2, since the network was sparser, and the location of all nodes was more stable than in Topology 3.

TABLE IV
LAKE EXPERIMENT. HAMMING DISTANCES $x/y/z$ (x: BETWEEN $\boldsymbol{X}(k)$ AND $\boldsymbol{Y}(k)$, I.E., ALICE/BOB; y AND z: MAX AND MIN BETWEEN $\boldsymbol{X}(k)$ AND $\boldsymbol{Z}(k)$, I.E., EVE/ALICE) AND OBSERVATION DELAYS IN [S] FOR DIFFERENT TOPOLOGIES AND ROUNDS.

		Topology 1	Topology 2		Topology 3			
		Round 1	Round 1	Round 2	Round 1	Round 2	Round 3	Round 4
Topology 1	Round 1	0/0/0	5/5/2	5/4/2	4/4/1	4/6/2	4/4/2	6/6/1
Topology 2	Round 1	5/5/2	0/0/0	0/6/0	1/6/2	3/5/1	3/5/2	3/6/1
	Round 2	5/4/2	0/6/0	0/0/0	1/5/2	3/5/4	3/5/2	3/5/2
Topology 3	Round 1	4/4/1	1/6/2	1/5/2	0/0/0	4/6/0	4/2/0	4/6/0
	Round 2	4/6/2	3/5/1	3/5/4	4/6/0	0/0/0	0/6/0	2/4/0
	Round 3	4/4/2	3/5/2	3/5/2	4/2/0	0/6/0	0/0/0	2/6/0
	Round 4	6/6/1	3/6/1	3/5/2	4/6/0	2/4/0	2/6/0	0/0/0
Observation delays [s]		28.7	24.7	25.0	16.8	19.2	13.9	16.5

The ideal conditions to extract different bits across multiple rounds of our scheme entail sufficiently slow mobility (to avoid Alice and Bob observing different propagation delays) but sufficiently fast mobility such that propagation delays change by more than ρ across subsequent rounds. As even the slow drifting observed in our experiment was enough to obtain some differences in the retrieved key sequences, we argue that stronger currents, as observed in open sea waters, would increase these differences even further. These results strengthen our claim that a large number of secret bits can be extracted by performing multiple sequence generation sessions.

The last line of Table IV, lists the observation delays for all rounds. We remark that different rounds with the same topology are a few minutes apart. We observe that differences between the measured delays still appear, even though the nodes drifted only slightly across rounds.

Fig. 10b shows the Hamming distance between Alice and Bob's delay vectors $d_H(\mathbf{X}(k), \mathbf{Y}(k))$ as well as the distance between Alice and Eve's vectors $d_H(\mathbf{X}(k), \mathbf{Z}(k))$ across all rounds, for $\rho = 0.04$ s and $\rho = 0.06$ s. As for Fig. 10a, we rotate the identity of Eve across all nodes different from Alice and Bob and visualize results for Eve in terms of the maximum and minimum Hamming distance of Eve's and Alice's delay vectors. We observe that Alice and Bob mostly agree on the self-generated key with a small number of bit errors that can be eventually corrected using standard procedures as in [26]. The results also show that Eve obtains 3 to 6 mismatched bits, which would increase proportionally to the number of rounds. Hence, for the seven rounds, we conclude that it is possible for Alice and Bob to agree on a bit sequence in a distributed fashion while keeping the sequence secret to Eve. This result holds for the different vantage points that Eve could have been located at, and although Eve used graph localization techniques to reconstruct the network. Finally, we explore the results in terms of the Hamming distance as a functino of the number of nodes in the network. Average results over the 7 trials are shown in Fig. 11. We observe that the key agreement between Alice and Bob slightly improves for smaller number of nodes in the network. This is because the potential of Alice or Bob to miss a route increases with the network size. On the other hand, the results show no significant change in terms of Eve capabilities. This stems from the tradeoff between knowing the network's topology and understanding its structure. That is, while the probability of Eve to extract the network's topology increases as the number of nodes decreases, her ability to determine the structure of the network to predict the links' time-of-flight decreases as the number of nodes decreases, where the later is due to the decrease in information avaialble for running the MDS algorithm.

In our work, we make realistic assumptions about Alice and Bob's capabilities. Among these, time-synchronization between Alice and Bob may seem the hardest. However, we note that ρ increases the robustness of the scheme against time-synchronization errors. Moreover, recent sharp decreases in the price and energy consumption of atomic clocks make built-in time-synchronization a valid possibility. Even without this, the synchronization assumption can be relaxed by allowing a four-way packet exchange between Alice and Bob. This way, the key is based on ToF accumulated in both the uplink and downlink directions, at the cost of doubling the delay for the completion of the sequence generation process.

From our results, we observe that a partial agreement between the delay sequences generated by Alice and Bob may still occur. These imperfections are caused by ToF estimation errors, e.g., due to multipath or unexpected hardware/software delays, but can also occur when R-Ping or R-Rep packets from an entire route fail to arrive in time. In such cases, Alice and Bob will fix differences in their bit sequences through the information reconciliation process [3], [26].

C. Discussion

The above results analyze the performance of our approach in terms of the number of secret bits, the key agreement between Alice and Bob, as well as the disagreement about Eve's key. From the results, we observe two possible weaknesses. One pertains to the diversity of the available information, which provides longer keys by multiple sequence generation sessions. Based on the results in Table II, we observe that the diversity in the topology and the observation delay is limited when the nodes' move slowly within consecutive sequence generation sessions, while the agreement rate between Alice and Bob increases. A second diversity source is the network's

topology. Here, based on the results in Table IV, we observe that the diversity (in terms of the observation delay) increases with the number of available links (e.g., compare Topologies 2 and 3).

The second weakness in our scheme concerns the capability to hide the information used for key generation from Eve. From Fig. 8a, we observe that Eve's capability to discover the key increases when the network becomes more connected. That is, our scheme works better in sparser networks. The penalty in denser networks can be mitigated by our mechanism to hide the number of nodes by using random IDs for the intermediate relay nodes. Yet, since underwater networks can be small, Eve may be able to overcome this obfuscation in some scenarios via trial-and-error attempts over its graph localization loss function.

In terms of robustness, the sensitivity of the proposed technique to the choice of both the quantization step, (ρ) and the number of routes (N_c), show that choosing an exceedingly large ρ and an exceedingly small N_c yields a smaller number of secret bits and thus a higher risk of exposing the key to Eve. Instead, a small ρ and large N_c may lead to key disagreements between Alice and Bob. Here, the user is expected to have bounds for the mobility of the nodes during the R-Ping–R-Rep exchange for a proper choice of these parameters.

VIII. Conclusions

We presented a scheme that leverages the propagation delay and the expected sparse topology in UWANs as a source of secret randomness. Our method builds upon the end-to-end delays observed by transmitting back and forth through different multihop routes between two nodes Alice and Bob, in order to generate a key that remains secret to an eavesdropping attacker Eve. By accumulating hardware/software delays along each route, Alice and Bob are able to measure the route's propagation delay and to autonomously-generate a common key by quantizing the estimated ToF accumulated over some intelligently chosen routes. Our simulation and experimental results show that our proposed scheme is secure against a passive eavesdropper, deployed randomly across the communications coverage area and attempting to reconstruct the network topology by means of graph localization. In particular, about 3.5 real secret key bits per packet exchange can be obtained even from small networks of 4 nodes. Our results also show that the proposed scheme is robust to nodes' mobility and to multipath-induced errors. Future work will explore more sources of randomness to extend the number of secret bits obtained from each packet exchange.

References

[1] G. Han, J. Jiang, N. Sun, and L. Shu, "Secure communication for underwater acoustic sensor networks," *IEEE journal of Mobile Computing*, vol. 53, no. 8, pp. 54–60, Aug. 2015.

[2] E. Souza, H. C. Wong, I. Cunha, . Cunha, L. F. M. Vieira, and L. B. Oliveira, "End-to-end authentication in under-water sensor networks," in *Proc. IEEE ISCC*, Split, Croatia, Jul. 2013.

[3] M. Bloch and J. Barros, *Physical-layer security: from information theory to security engineering*. Cambridge University Press, 2011.

[4] G. Dini and A. Lo Duca, "A secure communication suite for underwater acoustic sensor networks," *MDPI Sensors*, vol. 12, pp. 15 133–15 158, 2012. [Online]. Available: http://dx.doi.org/10.3390/s121115133

[5] I. Dokmanic, R. Parhizkar, J. Ranieri, and M. Vetterli, "Euclidean distance matrices: essential theory, algorithms, and applications," *IEEE Signal Process. Mag.*, vol. 32, no. 6, pp. 12–30, Jun. 2015.

[6] A. D. Wyner, "The wire-tap channel," *The Bell System Technical Journal*, vol. 54, no. 8, pp. 1355–1387, 1975.

[7] U. Maurer, "Secret key agreement by public discussion from common information," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 733–742, 1993.

[8] R. Ahlswede and I. Csiszar, "Common randomness in information theory and cryptography. I. Secret sharing," *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 1121–1132, 1993.

[9] Y. E. H. Shehadeh and D. Hogrefe, "A survey on secret key generation mechanisms on the physical layer in wireless networks," *Security and Commun. Networks*, vol. 8, no. 2, pp. 332–341, Feb. 2015.

[10] T. Wang, Y. Liu, and A. V. Vasilakos, "Survey on channel reciprocity based key establishment techniques for wireless systems," *Wireless Networks*, vol. 21, no. 6, pp. 1835–1846, 2015.

[11] E. Jorswieck, S. Tomasin, and A. Sezgin, "Broadcasting into the uncertainty: Authentication and confidentiality by physical-layer processing," *Proceedings of the IEEE*, vol. 103, no. 10, pp. 1702–1724, 2015.

[12] G. Li, C. Sun, J. Zhang, E. Jorswieck, B. Xiao, and A. Hu, "Physical layer key generation in 5G and beyond wireless communications: Challenges and opportunities," *Entropy*, vol. 21, no. 5, 2019.

[13] H. Kulhandjian, T. Melodia, and D. Koutsonikolas, "Securing underwater acoustic communications through analog network coding," in *Proc. IEEE SECON*, Singapore, Jun. 2014.

[14] L. Xiao, Q. Li, T. Chen, E. Cheng, and H. Dai, "Jamming games in underwater sensor networks with reinforcement learning," in *Proc. IEEE GLOBECOM*, San Diego, CA, Dec. 2015.

[15] Y. Liu, J. Jing, and J. Yang, "Secure underwater acoustic communication based on a robust key generation scheme," in *Proc. ICSP*, Beijing, China, Oct. 2008.

[16] R. Diamant, P. Casari, and S. Tomasin, "Cooperative authentication in underwater acoustic sensor networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 954–968, Feb. 2019.

[17] G. Leus and P. A. van Walree, "Multiband OFDM for covert acoustic communications," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 9, pp. 1662–1673, Dec. 2008.

[18] Y. Huang, S. Zhou, Z. Shi, and L. Lai, "Channel frequency response-based secret key generation in underwater acoustic systems," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 5875–5888, Sep. 2016.

[19] Y. Luo, L. Pu, Z. Peng, and Z. Shi, "RSS-based secret key generation in underwater acoustic networks: advantages, challenges, and performance improvements," *IEEE Commun. Mag.*, vol. 54, no. 2, pp. 32–38, Feb. 2016.

[20] Y. Huang, S. Zhou, Z. Shi, and L. Lai, "Experimental study of secret key generation in underwater acoustic channels," in *Proc. Asilomar Conf. on Signals, Systems and Computers*, Nov. 2014, pp. 323–327.

[21] ——, "Channel frequency response-based secret key generation in underwater acoustic systems," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 5875–5888, Sept. 2016.

[22] T. S. N. Murthy, G. S. Reddyka, and K. Padmaraju, "Adaptive secret key generation in underwater acoustic system," in *Proc. IEEE ICPCSI*, Sep. 2017, pp. 698–702.

[23] Z. Shen, J. Liu, and Q. Han, "A local pilot auxiliary key generation scheme for secure underwater acoustic communication," *Information Sciences*, vol. 473, pp. 1–12, 2019.

[24] K. Pelekanakis, C. M. G. Gussen, R. Petroccia, and J. Alves, "Towards physical layer cryptography for underwater acoustic networking," in *Proc. UACE*, Hersonissos, Crete, Greece, 2019.

[25] K. Pelekanakis, S. Yıldırım, G. Sklivanitis, R. Petroccia, J. Alves, and D. Pados, "Physical layer security against an informed eavesdropper in underwater acoustic channels: Feature extraction and quantization," in *Proc. UCOMMS*, 2021, pp. 1–5.

[26] G. Sklivanitis, K. Pelekanakis, S. Yıldırım, R. Petroccia, J. Alves, and D. Pados, "Physical layer security against an informed eavesdropper in underwater acoustic channels: Reconciliation and privacy amplification," in *Proc. UCOMMS*, 2021, pp. 1–5.

[27] K. Pelekanakis, C. M. G. Gussen, R. Petroccia, and J. Alves, "Robust channel parameters for crypto key generation in underwater acoustic systems," in *Proc. MTS/IEEE OCEANS*, Seattle, WA, Oct. 2019, pp. 1–7.

[28] C. Petrioli, G. Saturni, and D. Spaccini, "Feasibility study for authenticated key exchange protocols on underwater acoustic sensor networks," in *Proc. ACM WUWNet*, 2019, pp. 1–5.

[29] M. Xu, Y. Fan, and L. Liu, "Multi-party secret key generation over underwater acoustic channels," *IEEE Wireless Commun. Lett.*, vol. 9, no. 7, pp. 1075–1079, Jul. 2020.

[30] S. Zhang, X. Du, and X. Liu, "A secure remote mutual authentication scheme based on chaotic map for underwater acoustic networks," *IEEE Access*, vol. 8, pp. 48 285–48 298, 2020.

[31] X. Yu, H. Chen, and L. Xie, "A secure communication protocol between sensor nodes and sink node in underwater acoustic sensor networks," in *Proc. IEEE ICAICA*, 2021, pp. 279–283.

[32] L. Xu and T. Xu, *Digital Underwater Acoustic Communications*. Elsevier Science, 2016.

[33] J. Liu, Z. Wang, M. Zuba, Z. Peng, J. Cui, and S. Zhou, "DA-SYNC: A doppler assisted time synchronization scheme for mobile underwater sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 3, pp. 582–595, Mar. 2014.

[34] R. Shams, P. Otero, M. Aamir, and F. H. Khan, "Joint algorithm for multi-hop localization and time synchronization in underwater sensors networks using single anchor," *IEEE Access*, vol. 9, pp. 27 945–27 958, 2021.

[35] R. Otnes, J. Locke, A. Komulainen, S. Blouin, D. Clark, H. Austad, and J. Eastwood, "Dflood network protocol over commercial modems," in *Proc. UCOMMS*, 2018, pp. 1–5.

[36] R. Petroccia and J. Alves, "A hybrid routing protocol for underwater acoustic networks," in *Proc. Med-Hoc-Net*, 2018, pp. 1–8.

[37] G. Toso, I. Calabrese, P. Casari, and M. Zorzi, "RECORDS: A remote control framework for underwater networks," in *Proc. Med-Hoc-Net*, 2014, pp. 111–118.

[38] P. Casari and M. Rossi and M. Zorzi, "Towards optimal broadcasting policies for HARQ based on fountain codes in underwater networks," in *Proc. IEEE/IFIP WONS*, Jan. 2008, pp. 11–19.

[39] E. Isufi, H. Dol, and G. Leus, "Advanced flooding-based routing protocols for underwater sensor networks," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 52, May 2016.

[40] J. Potter, J. Alves, D. Green, G. Zappa, I. Nissen, and K. McCoy, "The JANUS underwater communications standard," in *Proc. UCOMMS*, Sestri Levante, Italy, Sep. 2014.

[41] E. Dubrovinskaya, P. Casari, and R. Diamant, "Bathymetry-aided underwater acoustic localization using a single passive receiver," *J. Acoustic Soc. Am.*, vol. 146, no. 6, pp. 4774–4789, Dec. 2019.

[42] R. Diamant, R. Francescon, and M. Zorzi, "A graph localization approach to assist a diver-in-distress," in *Proc. WPNC*, 2017, pp. 1–6.

[43] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY, USA: Wiley-Interscience, 2006.

[44] R. Diamant, H.-P. Tan, and L. Lampe, "LOS and NLOS classification for underwater acoustic localization," *IEEE Trans. Mobile Comput.*, vol. 13, no. 2, pp. 311–323, 2012.

[45] M. Porter, "Ocean acoustics library," http://oalib.hlsresearch.com.

[46] D. Mirza and C. Schurgers, "Collaborative localization for fleets of underwater drifters," in *Proc. MTS/IEEE OCEANS*, 2007, pp. 1–6.

[47] J. S. Jaffe, P. J. S. Franks, P. L. D. Roberts, D. Mirza, C. Schurgers, R. Kastner, and A. Boch, "A swarm of autonomous miniature underwater robot drifters for exploring submesoscale ocean dynamics," *Nature Commun.*, vol. 8, 2017.

[48] S. Subbaraya, A. Breitenmoser, A. Molchanov, J. Muller, C. Oberg, D. A. Caron, and G. S. Sukhatme, "Circling the seas: Design of lagrangian drifters for ocean monitoring," *IEEE Robot. Autom. Mag.*, vol. 23, no. 4, pp. 42–53, 2016.

[49] N. A. Cruz, B. M. Ferreira, O. Kebkal, A. C. Matos, C. Petrioli, R. Petroccia, and D. Spaccini, "Investigation of underwater acoustic networking enabling the cooperative operation of multiple heterogeneous vehicles," *Marine Technology Society Journal*, vol. 47, no. 2, pp. 43–58, mar 2013. [Online]. Available: https://doi.org/10.4031%2Fmtsj.47.2.4

[50] R. Costanzi, D. Fenucci, V. Manzar, M. Micheli, L. Morlando, D. Natale, M. Stifani, A. Tesei, and A. Caiti, "At-sea NATO operational experimentation with interoperable underwater assets using different robotic middlewares," in *Proc. Int. Conf. Ship and Maritime Research*, 2018.

[51] Y. Zhang, W. Feng, P. Pei, Y. Xu, and W. Xu, "Sea experiment for mobile underwater acoustic networks," in *Proc. ACM WUWNet*, 2019.

[52] J. Catipovic and S. Etchemendy, "Development of underwater acoustic modems and networks," *Oceanography*, vol. 6, no. 3, pp. 112–119, 1993. [Online]. Available: http://www.jstor.org/stable/43924652

[53] E. M. Sozer, M. Stojanovic, and J. G. Proakis, "Underwater acoustic networks," *IEEE J. Ocean. Eng.*, vol. 25, no. 1, pp. 72–83, Jan. 2000.

[54] N. Mohamed, I. Jawhar, J. Al-Jaroodi, and L. Zhang, "Monitoring underwater pipelines using sensor networks," in *Proc. IEEE HPCC*, 2010, pp. 346–353.

[55] W. M. Jassim and A. E. Abdelkareem, "Performance enhancement of oil pipeline monitoring for underwater wireless sensor network," in *Proc. CSASE*, 2020, pp. 38–43.

[56] EvoLogics GmbH, "S2C R 18/34 underwater acoustic modem," last visited: July 2021. [Online]. Available: https://evologics.de/acoustic-modem/18-34/r-serie