A Secure Cross-Layer Communication Stack for Underwater Acoustic Networks

Davide Tronchin[‡], Roberto Francescon[⋆], Filippo Campagnaro[‡⋆], Alberto Signori[‡], Roberto Petroccia[♯], Konstantinos Pelekanakis[♯], Pietro Paglierani[♯], João Alves[♯] Michele Zorzi[‡⋆§]

[‡] Department of Information Engineering, University of Padova, via Gradenigo 6/B, 35131 Padova, Italy
[§] Consorzio Futuro in Ricerca, via Saragat 1, 44122 Ferrara, Italy
[♯] NATO STO Centre for Maritime Research and Experimentation, Viale S. Bartolomeo, 400, 19126 La Spezia, Italy
[⋆] Wireless and More srl, Via della Croce Rossa 112, 35129 Padova, Italy
[‡]{tronchin,signoria,campagn1,zorzi}@dei.unipd.it,
[♯]{roberto.petroccia,konstantinos.pelekanakis,pietro.paglierani,joao.alves}@cmre.nato.int,
[⋆]{roberto.francescon,filippo.campagnaro,michele.zorzi}@wirelessandmore.it

***Abstract*—Underwater Acoustic Networks (UANs) have long been recognized as an instrumental technology in various fields, from ocean monitoring to defense settings. Their security, though, has been scarcely investigated despite the strategic areas involved and the intrinsic vulnerability due to the broadcast nature of the wireless medium. In this work, we focus on attacks for which the attacker has partial or total knowledge of the network protocol stack. Our strategy uses a `watchdog` layer that allows upper layers to gather knowledge of overheard packets. In addition, a reputation system that is able to label nodes as trustful or suspicious is analyzed and evaluated via simulations. The proposed security mechanism has been implemented in the DESERT Underwater framework and a simulation study is conducted to validate the effectiveness of the proposed solution against resource exhaustion and sinkhole attacks.**

I. Introduction

UANs have widely proven their effectiveness in many fields: coastal monitoring, ocean rescue, underwater facilities maintenance and maritime defense and many others. The broadcast nature of the acoustic medium is an enabling factor for the agility and the pervasiveness required by many advanced underwater applications [1]; however, this can be easily turned into a vulnerability by a malicious node, either compromised or introduced by an attacker, which may be able to exploit the ability to receive and send the packets that have been originated at the legitimate nodes of the network. Security aspects in UANs have not been widely investigated; however, works such as [2] and [3] investigate the security techniques available in radio networks that are applicable to the UANs context and propose a security framework able to exploit them.

An extensive classification of Denial Of Service (DoS) attacks in radio Wireless Sensor Networks (WSNs), along with the most common defensive techniques, is presented in [4] and [5]. Some of the DoS attacks that are performed in terrestrial radio networks are possible also in UANs. While some of them do not require knowledge of the protocol stack employed, such as jamming [6]–[8] or basic replay attacks [9], others exploit this knowledge to perform more sophisticated attacks such as sybil, wormhole and the sinkhole attack [10], [11]. Their countermeasures are often impossible to translate to the underwater acoustic domain, as its characteristics diverge too much from those of terrestrial radio, and therefore new studies and investigations are needed. Indeed, compared to wireless terrestrial communications, UANs are characterized by strong multipath, high packet error rate, long propagation delay and frequency-selective noise [12], as well as the lack of a standard infrastructure for public key and certificate exchange. In this context, countermeasures that are based on time validity of keys, certificates, freshness indices, that in general introduce a large overhead, cannot be directly applied to UANs.

Countermeasures based on observations of the neighbor nodes' behavior [13], instead, can be easily applied thanks to the broadcast nature of the underwater acoustic channel, as they require low computational power and introduce a small overhead to the communication. In this paper, we select two of the most common DoS attacks in radio WSN: resource exhaustion and sinkhole attacks. We present a simulation study and a general countermeasure based on a `watchdog` layer able to overhear the packets transmitted by the neighbor nodes. A reputation system based on the analyzed behavior is then applied to identify possible attackers and exclude them from the network.

In Section II we present and describe types of attacks that are common in WSNs and select those that are considered in this work, along with selecting the protocols to disrupt in the simulations. In Section III we propose countermeasures for each of these attacks and elaborate on the details for the selected scenarios. In Section IV we illustrate the framework in which the simulations of both the attacks and their countermeasures are performed, and in Section V we present the simulation results. Finally, in Section VI we draw our concluding remarks.

This work was supported in part by the NATO Allied Command Transformation (ACT) Future Solutions Branch under the Autonomous Security Network Programme and the Office of Naval Research Global under grant no. N62909-17-1-2093.

II. Protocol-aware attacks and countermeasures

The attacks we investigate and analyze in this work fall under the categories of *resource exhaustion* and *sinkhole* attacks. This is a generic terminology that accounts for a broad array of different attacks, although they have similarities in the strategy, within the same group.

Resource exhaustion attacks, in particular, include a wide range of attacks in which the malicious node tries to deplete some resources that are necessary for the attacked nodes to operate. In [14] the authors define a resource exhaustion vulnerability as a specific type of fault that causes the consumption or allocation of a resource in an undefined or unnecessary way, or the failure to release it when no longer needed, eventually causing its depletion; they simulate a wide array of attacks for the Domain Name System (DNS) in order to discover vulnerabilities. From the survey performed in [15], it emerges that the limited resources available at the sensor nodes are often subject to attacks that try to exploit all these weaknesses. Channel access and availability play an important role in these types of attacks and on the ability of an attacker to disrupt normal operation of the network. Among the proposed taxonomy of defenses, the watchdog type of countermeasures emerge as one of the most effective and lightweight.

The second type of attack we consider is the sinkhole attack. In this type of attack, the malicious node tries to convince the nodes of the network to route the traffic though itself. Then it can perform dropping or more sophisticated traffic analysis. From this attack, more complex strategies can be mounted such as the *wormhole* attack [16], [17]. The survey in [18] remarks the necessity to design routing protocols with security in mind and argues that this is something rarely addressed. The work performs a survey of some of the most common WSN routing protocols, common attacks and countermeasures: the conclusion is that a defense mechanism against the sinkhole attack is unlikely to be designed, and that the only possibility is to design the routing protocols around the sinkhole attack countermeasure. The countermeasure taken in [19], which is one of the most common in the literature, is quite expensive in terms of overhead as it is based on specific signaling messages dedicated to the attacker detection. Specifically, it proposes an algorithm that relies on the presence of a data collection point (sink node) to detect a possible ongoing sinkhole attack involving flooding the network and retrieving information.

A. Resource Exhaustion Attack on UWPOLLING

To test the resource exhaustion attack and its defense, we chose UWPOLLING, a polling-based Medium Access Control (MAC) protocol [20] that can be used for data muling applications, i.e., in networks where a mobile node, such as an Autonomous Underwater Vehicle (AUV) or a surface vessel, acts as the sink of the network, and collects data from submerged sensor nodes. In each UWPOLLING cycle, the sink first sends a trigger packet to the sensor nodes and then waits for the sensor nodes to answer with a probe packet, that contains information on how much data each node needs to transmit to the sink. Then, the sink sends a poll packet to assign each sensor node the time interval within which that node can transmit its own data packets. The sink selects the next node to be polled according to a fair policy, i.e., it gives higher priority to nodes from which it received less data. Afterwards, the sink starts a new UWPOLLING cycle with a new trigger packet.

We simulated two types of attacks. In the first type, the attacker is only able to store the overheard packets and replay a subset of them; in detail, the attacker is able to discriminate the different types of signaling packets and performs the attack by replaying: trigger packets, poll packet and probe packets. In the second type of attack, a smart attacker is able to generate legitimate signaling packets; in this case, the attacker's goal is to exhaust the channel access allocation by asking the sink the permission to transmit a large amount of packets, without actually transmitting them; therefore, the sink always polls the attacker first, due to its fair policy, thus resulting in less time for the other nodes in range to transmit their data.

We assume two types of attacks for this protocol:

- a malicious node replaying only a selected subset of the recorded packets;
- a malicious node generating deceptive legitimate packets that cause the sink to assign excessive resources to the attacker.

B. Sinkhole Attack on SUN protocol

To test the *sinkhole* attack we chose the SUN protocol [21]. This is a Dynamic Source Routing (DSR) protocol based on hop-count for route determination and includes two types of nodes: a sink node that collects information and normal nodes that generate this information. The basic mechanism of SUN is described below:

1) the sink sends probe packets to allow in-range nodes to be aware of being one-hop away from the sink;
2) nodes that receive probe packets, called *end-nodes*, have now a route to the sink node, of length one hop;
3) a node that has data to send, and does not have any route to the sink broadcasts a path request;
4) a node receiving a path request can perform one of the following actions:
    a) if it has no route to the sink, it adds its own address to the request packet and broadcasts again the packet;
    b) if it is an end-node, it adds its own address to the packet and sends back a path establishment reply, using the reverse route;
    c) if it has a route to the sink but it is not an end-node, it proceeds as in case (a);
5) the path establishment reply packet, containing all the nodes that the corresponding request passed through, reaches the source node, allowing it to have a route to the sink node.

The malicious node will always advertise itself as an end-node, i.e., as being one hop away from the sink node. Then, the attacker will drop either all or some of the packets received

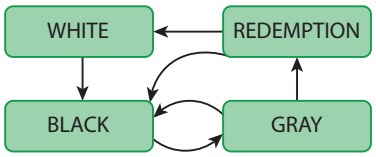

Fig. 1. State machine of the reputation system.

by the other nodes, instead of forwarding them: in this work, the effect of the attack is analyzed by varying the position of the malicious node and the percentage of packets that it drops.

III. Defense framework and strategies

The defense strategy we present in this work develops upon two fundamental mechanisms: a `watchdog` layer, able to overhear and send the overheard data to the upper layers, and a reputation mechanism, able to label nodes as *suspicious* or *trustful* and thus to choose which ones to rely on.

The `watchdog` layer is placed between the MAC and the physical layer and is able to overhear all communications of the nodes in range. Via cross-layer interaction, this layer can notify all other layers of the stack when it overhears a packet and is the base of the security architecture employed. The interested layers collect this information and use it in their specific defense strategy.

In order to detect and counteract the ongoing attack, a reputation system which makes use of the `watchdog` security layer and of the shared cross-layer information is employed. Each protocol of the communication stack implementing the required security mechanism needs to perform the following operations:

1) check which node transmitted the overheard packet received via cross-layer;
2) decide, according to its protocol rules, whether that node is behaving in a good (`G`) or in a bad (`B`) way;
3) report the node's behavior to the reputation system.

We want to highlight that the good (`G`) and bad (`B`) ways are protocol dependent, and the reputation system only needs to know the final results of the behavior, i.e., `G` or `B`. Figure 1 shows the state machine employed by the reputation system illustrated in this work. Then the reputation system, based on its implementation, decides whether that node should be trusted or not by using white, grey, redemption and black lists. The transition between white, gray, redemption and black is generic and can change depending on the implementation of the reputation system. We present the implementation we used for the state transitions as follows.

- Initially, all nodes are in the white list and have a total trust $s_i = S_{max}$, that is the maximum trust score.
- Each time the behavior of a node x is `B`, s_x is decremented, and each time the node behavior is `G` for n_{res} consecutive times, s_x is restored to the maximum value S_{max}.
- If s_x becomes 0, node x is blacklisted.
- Depending on the layer rules, after a certain event (e.g., after a timeout elapses, or after 10 selections of other nodes as the next hop of the network), a blacklisted node x is moved to the gray list in order to give x a second chance.
- A node x in the gray list is moved to the black list as soon as it behaves in `B` way, instead if its behavior is `G` for n_{gr} times it is moved to the redemption list
- A node x in the redemption list has a total score of $s_x = S_{max}/2$: each time its behavior is `B`, s_x is decremented, and if s_x becomes 0, x is blacklisted. Instead, if its behavior is `G` for n_{rw} times, it is moved to the white list and gains back a total score of S_{max}.

A. Resource Exhaustion Countermeasures

To counteract the first attacker strategy based on the replay of recorded packets, a security mechanism based on the HASH freshness index is applied, as described in [9]: a security layer computes the XOR operation between the HASH of the packet generation time and the HASH of the network address of the source node, for a total size of 4 Bytes, and inserts it in the packet header, for the receiver to certify its validity. For the second type of attack, based on the generation of legitimate packets, two complementary and subsequent phases of the countermeasure system are employed. The first phase is a preliminary check used to immediately exclude from the network those nodes that ask to transmit too many data packets (thus, with a highly suspicious behavior); specifically, when the amount of time to be allocated to a single node is below a given threshold $T_{tx,ths}$ the packet is directly inserted in the poll list, whereas, if the amount of time exceeds the threshold, the node is either not inserted in the poll list (BAN-NODE mechanism), or inserted but reducing the amount of allocated time for it (POLL-NODE mechanism). The second phase, that consists in the actual countermeasure, is based on the reputation system presented in Section III and comes into play when the first countermeasure is not applicable, i.e., when the number of packets the node is asking permission to transmit is not immediately suspicious. Indeed, a smarter attacker can bypass the former countermeasure by reducing the number of packets it asks to transmit. The reputation system gives a score to each node and reduces it when there is a difference between the intended transmissions and the actual packets transmitted. Eventually, the malicious node is blacklisted by the reputation system and no longer considered in the polling phase.

B. Sinkhole Countermeasures

The defense mechanism is based on overheard packets: after a node asks a neighbor to forward a packet, it observes whether that neighbor forwards that packet within a certain timeout by using the `watchdog` layer. If the packet is not overheard, the node assumes that the neighbor dropped that packet and decreases the trust score of this neighbor. Conversely, if the source node overhears a correctly forwarded packet, it increases the score of the relay node. The increase/decrease of reputation score operation is performed using the reputation mechanism presented in Section III. When a source node needs to send data, there could be two cases: in case the node has no route to the sink it begins the path establishment process that

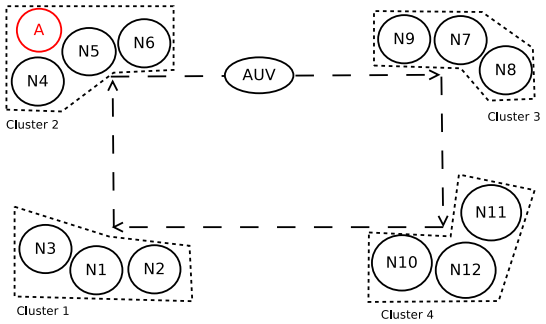

Fig. 2. POLLING attack simulation topology.

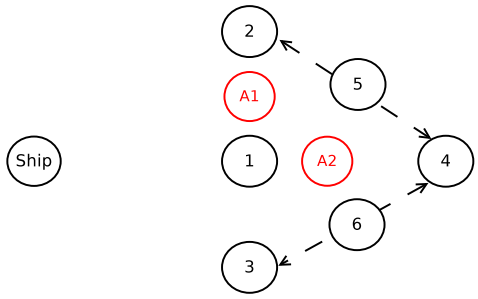

Fig. 3. Sinkhole attack simulation topology.

will be analyzed later; in case the node has a route to the sink, it verifies the trust of the first hop: if the node is blacklisted (i.e., it has a bad reputation), the route is discarded and another path establishment process is started. The path establishment process begins with a path search packet to which, hopefully, a number of nodes reply with a path answer packet. Once receiving the reply messages, the source node will verify the trust of the replying nodes and discard all the routes coming from blacklisted nodes. Among the remaining ones, the path with the minimum number of hops is selected. Otherwise, if no trusted route is received within a fixed timeout, a new path establishment process is started.

IV. Simulation scenarios and system settings

The DESERT Underwater simulator [22] is used to simulate the two selected types of attacks and their countermeasures. The characteristics of both the devices and the environment are slightly different between the resource exhaustion attack and the sinkhole attack simulations, to better highlight the key behaviors.

In the simulation of the *resource exhaustion* attack, all nodes are equipped with low frequency acoustic modems operating in the 7-17 kHz bandwidth, their transmission rate is 500 bit/s and their simulated range is 4 km.

In the first scenario the attacker is not able to create legitimate packets but can discriminate their type, namely: `data`, `trigger`, `probe` and `poll` packet. We simulate the disruption of the POLLING protocol replaying, separately, `trigger`, `probe` and `poll` packets: the topology is presented in Figure 2. The nodes are deployed in 4 clusters composed by 3 legitimate nodes each. In the second cluster an attacker is placed close to the legitimate nodes, in order to be able to record and repeat the signaling packets generated within its cluster or by the AUV. This setup allows to quickly compare the behavior of the attacked cluster with the clusters close to it and the one not reached by the attacker. The AUV moves at a constant speed of 1.67 m/s in the network following a square path, moving among the clusters and collecting data from the nodes. In this configuration, all nodes generate a packet every 120 seconds with a size of 60 bytes. We considered 3 cases in which the attacker retransmits packets of different sizes: in the first case, only `trigger` packets are retransmitted; in the second case, only `poll` packets are retransmitted; finally, in the third case, only `probe` packets are retransmitted.

In a second scenario, a smarter attacker is able to forge legitimate packets. In this case, the attacker's goal is to exhaust the channel access allocation by asking the sink permission to transmit a large number of packets, and never transmitting them. Therefore, at each trigger cycle when the attacker is in range with the sink, the sink always polls the attacker first due to its fair policy, giving the other nodes in range less time to transmit their data.

For what concerns the simulation of the *sinkhole* attack, we test it on the SUN network protocol using the topology shown in Figure 3. The attacker, depicted in red, is first placed in position A1 and then in position A2 to inspect if the attacker's position affects the network in different ways. All the devices involved are operating in the 7-17 kHz frequency range, their bitrate is 500 bit/s and the communication range for each node is about 4 km, therefore nodes 1, 2 and 3 are in communication range with the sink while node nodes 4, 5 and 6 require the usage of relay nodes. Node 4 is in communication range with 1, 2 and 3. For nodes 5 and 6 (two AUVs), the link quality to reach the other nodes is impacted by the vehicles' mobility. As previously specified, the goal of the attacker is to attract the largest possible amount of traffic from the network: then, it can decide to drop some of this traffic.

TABLE I
Traffics of the nodes deployed in the scenario of Figure 3

Traffic	Generating Node	Packet size [bytes]	Generation period [s]
T1	1, 2, 3, 4	32	120
T2	5, 6	60	120
T3	5, 6	120	80

The nodes generate packets at a constant rate and size: the characteristics of the various types of traffic, generated by the nodes in the scenario in Figure 3, are detailed in Table I.

V. Results

A. Resource Exhaustion Attacks

We now analyze the impact of the *first resource exhaustion attack*, performed by a malicious node that replays signaling packets, on the throughput of the network as a function of the replay time T_{replay}, i.e., the average time between two consecutive attacker's transmissions.

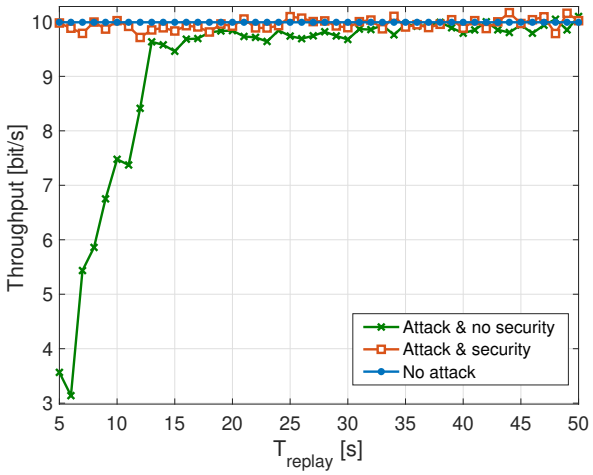

Fig. 4. Throughput of Cluster 2 when the malicious node repeats trigger packets

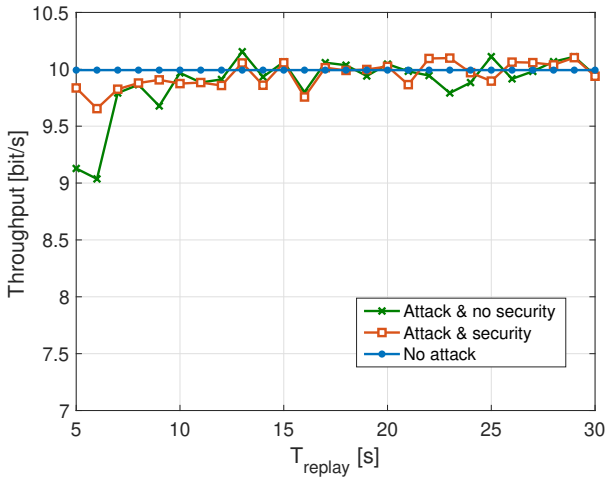

Fig. 6. Throughput of Cluster 2 when the malicious node repeats probe packets

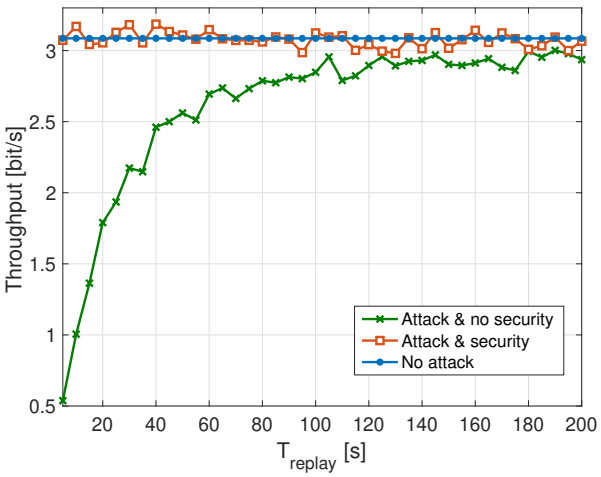

Fig. 5. Throughput of the attacked node when the malicious node repeats poll packets

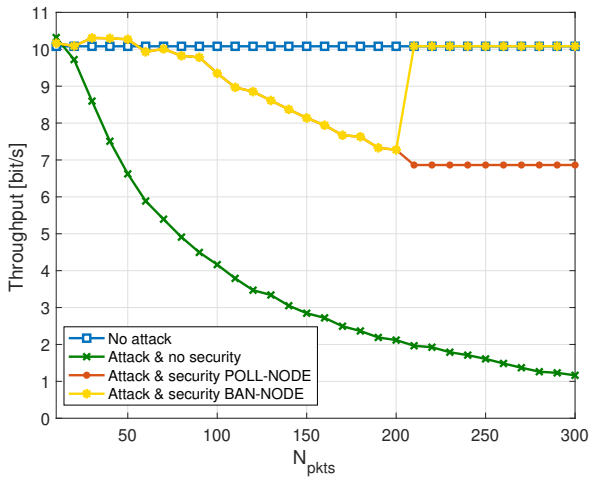

Fig. 7. Overall throughput of the second cluster for the resource exhaustion attack

Figure 4 shows the overall throughput, for Cluster 2 of the topology depicted in Figure 2, as a function of T_{replay}, when the attacker retransmits trigger packets. This type of attack has small effects for $T_{replay} > 15$ s (green line). Conversely, the throughput of Cluster 2 drops to 7 bit/s when $T_{replay} = 10$ s, and further decreases down to 3 bit/s when T_{replay} decreases to values < 10 s. When the security mechanism based on the HASH freshness index is enabled (red line), the throughput of the network is almost equivalent to the case without attack (blue line), confirming the effectiveness of this countermeasure.

Figure 5 depicts the throughput of the node affected by the periodic replay of the same poll packet. The poll packet is sent in unicast, therefore when a poll packet is replayed only one node is attacked, i.e., only the intended node of the original poll packet. Figure 5 shows that the throughput of the attacked node is affected even with a relatively high replay period, i.e., with $T_{replay} < 120$ s. As for the trigger packet attack, the security mechanism based on HASH freshness index completely mitigates the effectiveness of this attack.

Differently from the previous scenarios, the replay of a probe packet does not affect the performance of the network (Figure 6) except for small values of T_{replay}, where there is a small drop in the overall throughput. Also in this case, the HASH index provides a valid countermeasure, preventing the effects of the attack.

The effects of the *second resource exhaustion attack*, performed by a node able to generate legitimate probe packets, are presented in Figure 7. The figure shows the overall throughput of Cluster 2, i.e., the cluster where the attacker is located. The green line represents the overall throughput of Cluster 2 when the network is under attack and without defense. The

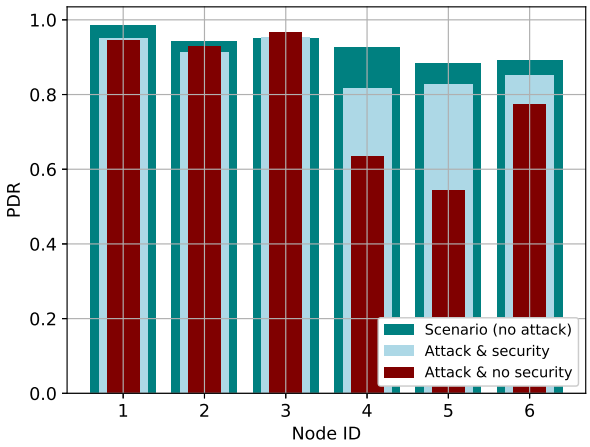

Fig. 8. Sinkhole attack: PDR with attacker in position A1

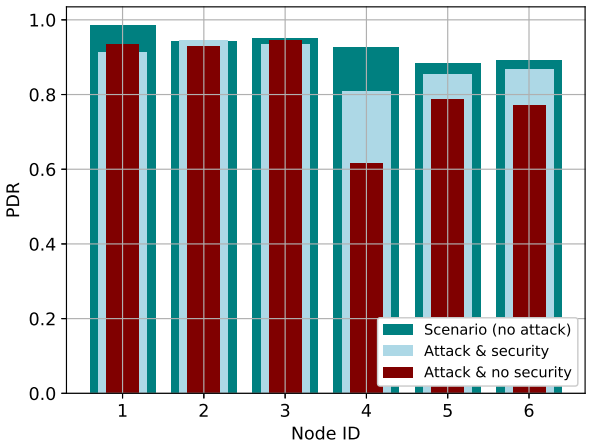

Fig. 9. Sinkhole attack: PDR with attacker in position A2

analysis has been done as a function of the number of packets the attacker claims to transmit in the probe packets (N_{pkts}). The larger the value of N_{pkts}, the higher the amount of time reserved to the attacker, and, therefore, the lower the time available for the other nodes to be served by the AUV. In this scenario, the overall throughput of the cluster is reduced by 30% when $N_{pkts} = 50$ and by 60% when $N_{pkts} = 100$. Using the security mechanisms, the drop in performance is limited. When $N_{pkts} > 200$, the amount of time that should be reserved to the attacker exceeds the time threshold $T_{tx,ths}$. If the BAN-NODE mechanism is employed (yellow line), the node is automatically excluded from the network. Indeed, the performance becomes comparable to the performance without an attacker (blue line). When the POLL-NODE mechanism is employed, there is a reduction in the overall throughput, since an amount of time equal to the threshold $T_{tx,ths}$ could be reserved for the attacker. Still, the effects of the attacker are mitigated thanks to the reputation system. Indeed, when no data packets are received from the attacker, its reputation is reduced as long as the node is inserted in the black list. From that point onward, except for the redemption attempts, the node is not considered and $T_{tx,ths}$ s are no longer reserved. When $N_{pkts} \leq 200$, the barrier does not come into play, and only the reputation system can prevent the malicious node from affecting the network performance. When $N_{pkts} \leq 100$, the performance of the network with the security mechanism is almost equivalent to the network without attack. For higher values of N_{pkts} there is a drop in performance, but still much lower than the drop without any security countermeasure. The drop with the security mechanism is due to the fact that the reputation system takes time to identify the malicious node and put it in the black list. In the meantime, the attacker still induces the AUV to reserve the channel to it, thus reducing the channel availability for the other nodes in the cluster.

B. Sinkhole Attack

Concerning the sinkhole attack against the SUN network protocol, Figure 8 and Figure 9 present the Packet Delivery Ratio (PDR) of the nodes in the network when the attacker is in position A1 and A2, respectively. In both plots the attacker drops 100% of the received packets. When the attacker is in position A1, the most affected nodes are node 4 (static node) and node 5 (the first AUV), whose PDR drops from 93% to 64%, and from 88% to 54%, respectively, when the network is under attack. With the countermeasure enabled, the PDR increases to 82% for node 4 and to 83% for node 5, thanks to the fact that the malicious node is detected and blacklisted by the reputation system. When the attacker is in position A2, the most affected node is node 4, whose PDR drops from 93% (without attack) down to 62% (ongoing attack). With the security enabled, the node's PDR increases to 81%. For both cases, the nodes most affected by the attack are those that are not in range with the sink, and need to find possible routes. The capability to detect the presence of the malicious node and avoid its selection as relay node has a significant impact on the data delivery. From these figures, it can be seen that some nodes improve their performance in case of attack and no countermeasure deployed: this happens because the network is loaded with heavy traffic, thus, nodes close to the sink have to both send their packets and relay packets for other nodes. When the attacker is in position A1, and drops most or all of the packets, its behavior is beneficial to those nodes whose traffic is mostly relayed for other nodes, namely node 1, node 2 and node 3: in the first two cases the PDR does not undergo a noticeable change, whereas, in case of node 3, it is even increased. When the attacker is in position A2, the PDR of the nodes close to the sink does not vary notably.

Figure 10 and Figure 11 show the PDR of the nodes most affected by the attack when the attacking node varies the percentage of dropped packets, respectively, in position A1 and position A2. When the attacker is in position A1 it is able to effectively draw traffic from both node 4 and node 5. Figure 10 shows that the defense mechanism is able to restore the PDR to values very close to the scenario without attack. Figure 11 shows that, when the attacker is in position A2, only node 4 is heavily affected by the attack, as node 5 is

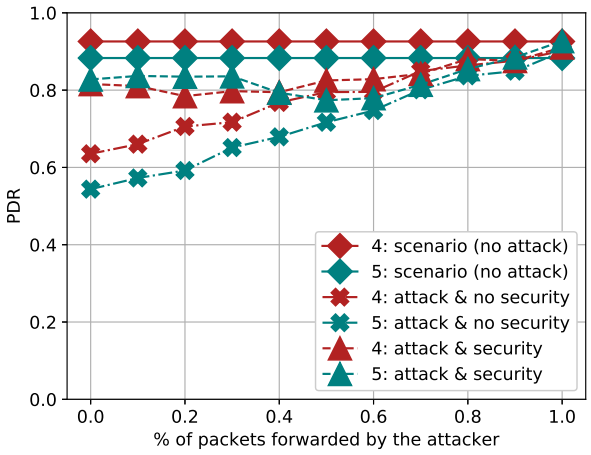

Fig. 10. Sinkhole attack: PDR of the most affected nodes when the attacker is in position A1

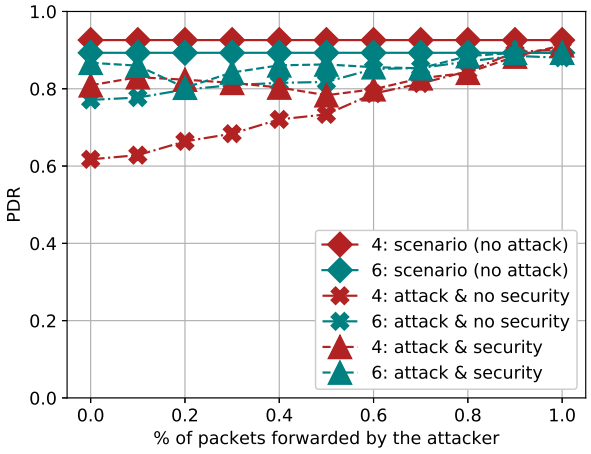

Fig. 11. Sinkhole attack: PDR of the most affected nodes when the attacker is in position A2

now able to obtain a valid route from node 2 more frequently. In addition, the defense mechanism restores the PDR to values close to the no-attack scenario. In both cases, it can be seen that the most challenging situation for the defense mechanism is when the dropped percentage is in mid-range. When the attacker is dropping all packets it is easier for the other nodes to detect the ongoing attack. Conversely, when the attacker relays most or all of the packets, it is harder for other nodes to detect the attack but the PDR is closer to no-attack values.

VI. Conclusions

The proposed defense mechanism, based on a `watchdog` layer and a *reputation system*, proved to be effective in counteracting the two types of attacks analyzed. In particular, it is able to prevent the attacked nodes from being excluded completely from the network, guaranteeing a minimum level of participation in the network communication. This includes defending against attacks that exploit knowledge of the communication protocol stack. Specifically, a countermeasure based on a packet freshness index has been proved as a valuable solution for replay attacks, whereas a reputation based system can limit the effect of sinkhole and resource exhaustion attacks. Furthermore, the defense framework, implemented in DESERT Underwater [22], proved to be very extensible and configurable, allowing to tailor the general structure to the attacks under analysis, in both the `watchdog` layer and the trust mechanism. We expect to further specialize the framework, implementing additional defense strategies specifically suited to new types of attacks and exploring nodes cooperation.

References

[1] M. Stojanovic, "Underwater wireless communications: Current achievements and research challenges," *IEEE Oceanic Engineering Society Newsletter*, vol. 41, no. 2, pp. 1–5, 2006.

[2] C. Lal, R. Petroccia, K. Pelekanakis, M. Conti, and J. Alves, "Towards the development of secure underwater acoustic networks," *IEEE Journal of Oceanic Engineering*, vol. 42, no. 4, pp. 1075–1087, October 2017.

[3] G. Yang, L. Dai, and Z. Wei, "Challenges, threats, security issues and new trends of underwater wireless sensor networks," *Sensors*, vol. 18, no. 11, p. 3907, Nov. 2018.

[4] A. D. Wood and J. A. Stankovic, "A taxonomy for denial-of-service attacks in wireless sensor networks," *Handbook of sensor networks: compact wireless and wired sensing systems*, pp. 739–763, 2004.

[5] D. R. Raymond and S. F. Midkiff, "Denial-of-service in wireless sensor networks: Attacks and defenses," *IEEE Pervasive Computing*, vol. 7, no. 1, pp. 74–81, Jan. 2008.

[6] M. M. Zuba, Z. Shi, Z. Peng, and J.-H. Cui, "Launching denial-of-service jamming attacks in underwater sensor networks," in *Proceedings of the Sixth ACM International Workshop on Underwater Networks*, ser. WUWNet'11, Seattle, Washington, USA, 1–2 December 2011.

[7] M. Goetz, S. Azad, P. Casari, I. Nissen, and M. Zorzi, "Jamming-resistant multi-path routing for reliable intruder detection in underwater networks," in *Proceedings of the Sixth ACM International Workshop on Underwater Networks*, ser. WUWNet'11, Seattle, Washington, USA, 1–2 December 2011.

[8] A. Signori, F. Chiariotti, F. Campagnaro, and M. Zorzi, "A game-theoretic and experimental analysis of energy-depleting underwater jamming attacks," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9793–9804, Oct. 2020.

[9] F. Campagnaro, D. Tronchin, A. Signori, R. Petroccia, K. Pelekanakis, P. Paglierani, J. Alves, and M. Zorzi, "Replay-attack countermeasures for underwater acoustic networks," in *Proc. MTS/IEEE Oceans*, Virtual Conference, Oct. 2020.

[10] I. Krontiris, T. Giannetsos, and T. Dimitriou, "Launching a sinkhole attack in wireless sensor networks; the intruder side," in *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*. IEEE, 2008, pp. 526–531.

[11] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: analysis & defenses," in *Third international symposium on information processing in sensor networks, IPSN 2004*. IEEE, 2004, pp. 259–268.

[12] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," *ACM Mobile Comput. and Commun. Review*, vol. 11, no. 4, pp. 34–43, Oct. 2007.

[13] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, 2000, pp. 255–265.

[14] J. Antunes, N. F. Neves, and P. J. Veríssimo, "Detection and prediction of resource-exhaustion vulnerabilities," in *19th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2008, pp. 87–96.

[15] O. A. Osanaiye, A. S. Alfa, and G. P. Hancke, "Denial of service defence for resource availability in wireless sensor networks," *IEEE Access*, vol. 6, pp. 6975–7004, Jan. 2018.

[16] R. Zhang and Y. Zhang, "Wormhole-resilient secure neighbor discovery in underwater acoustic networks," in *Proceedings of 29th IEEE International Conference on Computer Communications*, ser. INFOCOM'10, San Diego, CA, USA, 15–19 March 2010, pp. 1–9.

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

[18] A. Mathew and J. S. Terence, "A survey on various detection techniques of sinkhole attacks in WSN," in *International Conference on Communication and Signal Processing (ICCSP)*. IEEE, 2017, pp. 1115–1119.

[19] E. C. Ngai, J. Liu, and M. R. Lyu, "On the intruder detection for sinkhole attack in wireless sensor networks," in *IEEE International Conference on Communications*, vol. 8. IEEE, 2006, pp. 3383–3389.

[20] A. Signori, F. Campagnaro, D. Zordan, F. Favaro, and M. Zorzi, "Underwater acoustic sensors data collection in the robotic vessels as-a-service project," in *Proc. MTS/IEEE Oceans*, Marseille, France, June 2019.

[21] G. Toso, R. Masiero, P. Casari, M. Komar, O. Kebkal, and M. Zorzi, "Revisiting source routing for underwater networking: The sun protocol," *IEEE Access*, vol. 6, pp. 1525–1541, Dec. 2018.

[22] "DESERT Underwater," last time accessed May. 2021. [Online]. Available: http://desert-underwater.dei.unipd.it/