Replay-Attack Countermeasures for Underwater Acoustic Networks

Filippo Campagnaro[‡], Davide Tronchin[‡], 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
[♯] NATO STO Centre for Maritime Research and Experimentation, Viale S. Bartolomeo, 400, 19126 La Spezia, Italy
[§] Consorzio Futuro in Ricerca, via Saragat 1, 44122 Ferrara, Italy
[‡]{campagn1,tronchin,signoria,zorzi}@dei.unipd.it,
[♯]{roberto.petroccia,konstantinos.pelekanakis,pietro.paglierani,joao.alves}@cmre.nato.int

***Abstract*—Security is an important service of Underwater Acoustic Networks (UANs). In this work, we investigate the impact of four different replay attacks and propose two counteracting methods. The first method is based on the observation of the packet generation timestamp and the second method uses the HASH value of the packet generation timestamp combined with the address of the source node. These two methods are implemented between the routing and Media Access Control (MAC) layers of the DESERT communications protocol stack. Our simulation results show that the proposed countermeasures almost neutralise the attacks for two different network topologies.**

I. Introduction

Underwater Acoustic Networks (UANs) have been recognized as an enabling technology for various applications in the maritime domain. The wireless nature of the acoustic medium makes UANs vulnerable to various malicious attacks, yet, limited consideration has been given to security challenges in this environment so far [1], [2]. Various types of Denial-of-Service (DoS) attacks can be conducted in UANs. Some of these attacks assume the ability of the malicious node to produce or manipulate legitimate messages, e.g., *Sinkhole* and *Wormhole* attacks. For other attacks, instead, the malicious node does not need to be able to generate any legitimate message to disrupt the network operations, e.g., jamming and basic replay attacks[1] Additionally, no sophisticated hardware or capable processing capability is required.

Although countermeasures to DoS attacks have been widely studied in the radio frequency domain [3], [4], only few solutions have been proposed for UANs, mainly focusing on jamming attacks [5]–[7], wormhole attacks [8], [9] and the usage of security tools to protect the integrity and confidentiality of the received messages [10]–[13]. In this work, we investigate countermeasures against a replay attack (Figure 1). In this type of attack, the malicious node records messages transmitted by legitimate nodes in the network and replays these messages.

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.

[1]In the replay attack considered in this paper a node can only receive and retransmit a packet as it is, without the ability to modify its content.

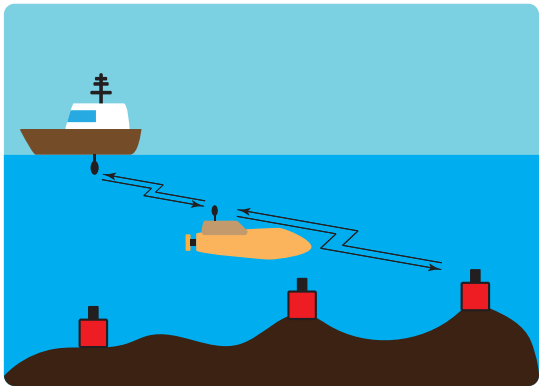

Fig. 1. Replay attack: an AUV, acting as a malicious node, records packets transmitted by the surrounding nodes and re-injects them into the network.

In this work, we assume the attacker knows the waveform used for the transmission, i.e., it is able to decode the packet, but without knowing the protocol stack used in the network, therefore with no capabilities to understand the content of the packets. The objective of the attacker is to waste the scarce network resources.

Different packet replay strategies can be used by the attacker, depending on its capabilities and on the selection of the packet(s) that will be replayed.

In this work, we analyze the effect of different replay attack strategies in a multi-hop UAN. Additionally, various countermeasures are proposed through the development of a network security layer with cross-layer capabilities, thus minimizing the overhead required for a secure communication. To validate and evaluate the proposed solution, a simulation study is conducted using the DESERT Underwater Framework [14] where all the various attacks and possible countermeasures are implemented.

In Section II, after reviewing the state of the art, we present all the configurations of the replay attack investigated in this paper, as well as the design of the security layer used as countermeasure for this type of attack. In Section III we describe the scenario and the network topologies used to test the attacks and the countermeasures, while in Section IV we

present the simulation results and therefore the evaluation of both the attacks and the security system. Finally, in Section V we draw our concluding remarks.

II. Replay Attacks and Countermeasures

A replay attack is an attempt to perform a malicious action by recording valid data transmissions and repeating or delaying them in order to impersonate a valid user in a network. The replay attacks can be classified in straight replays, where the packets are intended for the same destination but delayed, and deflections, where the packets are directed to other than the intended recipient, e.g., reflected back to the sender, or deflected to a third node [15][2]. Despite the fact that we are not considering any packet encryption, in this paper we assume that the malicious node only knows the waveform used for the transmission and is not aware of the protocols employed in the network, therefore it cannot modify a recorded message, but can only send it later in time. However, when this attack is performed by a mobile node in a multihop network, the attacker can still forward packets to nodes that are not neighbors of the transmitter. In the classical replay attack, the intruder records the data transmitted in the channel for a certain amount of time, and then replays the whole recorded signal as it is [16]. A smarter attacker can also identify the packets from the signal, decode them and decide whether to transmit all or a part of them, selected in order of arrival or chosen at random, one or multiple times [17].

Replay attack countermeasures for wireless terrestrial networks have been discussed for a long time in the literature. Traditional security methods (e.g., cryptography) do not provide complete protection against replay attacks [18]. Some attempts to use timestamp methods in the packets have provided some benefits [19]. The use of timestamps, however, could not be applicable in case of lack of synchronisation in the network [18], however, given the low bandwidth and the large packet delivery delay experienced in underwater acoustic networks, in our scenario a large validity period can be set, by overcoming the synchronization issue and thus indicating that a timestamp-based solution should be investigated. The authors in [19] provided an authentication protocol for preventing replay attacks. The protocol gave a mechanism to inspect the message freshness (e.g., serial number, timestamp). Nevertheless, this system requires the exchange of several messages for sharing the keys used for the authentication of the legitimate nodes, and may not be directly applicable to a UAN. The authors in [20] and [21] have studied the effects of replay attacks in secure ZigBee networks. They showed that ZigBee networks are vulnerable to replay attacks also when using encrypted payloads and a frame counter. Authors thus suggest a full timestamp scheme as replacement of the frame counter mechanism. As a first solution we propose a similar scheme, reduced in size and complexity to fit the underwater acoustic scenario. The authors in [22], instead, demonstrate that a protection for the replay attack based on the hash value of the bits of the packet outperforms the frame counter method. This strategy has the advantage that no additional bits need to be added to the packet, and fits well WiFi networks, where the packet header already includes a timestamp and the packet size is usually very large compared to the size of the packets transmitted in an acoustic networks. Indeed, applying this solution directly to the bits of the short packets transmitted in acoustic networks, where the packet header size is minimized and may not contain a timestamp, would lead to a high HASH collision probability. This problem can be mitigated with the addition of a timestamp to the bits used as input to the HASH function: our second countermeasure is based on this solution.

In this work, four types of replay attack strategies are analyzed, in order to inspect which defense mechanism best reacts against different attacks.

1) FIRST-PACKET: only the first packet detected by the malicious node is replayed with a given repetition time during the entire simulation. This is the simplest attack, as the attacker needs to record only one packet and then retransmit it repeatedly.
2) LAST-PACKET: only the last packet sensed by the malicious node is replayed with a given repetition time during the entire simulation. The identification of the presence of this attack is quite hard, as the malicious node always transmits fresh data.
3) MULTI-PACKET: all packets sensed by the malicious node are recorded and replayed only once. Based on the considered strategies, the amount of traffic injected by the MULTI-PACKET approach depends on the number of messages transmitted by the legitimate nodes, while the other attacks inject a fixed number of packets, independent of the network traffic.
4) HOLD-PACKET: all packets sensed by the malicious node are recorded. After a certain amount of time (e.g., one hour of recording), the attacker chooses at random and replays one of the recorded packets at a time [16].

All attacks aim to inject packets into the network, with the goal to fill the Medium Access Control (MAC) queues of the nodes, and, therefore, saturate the network.

In this paper, we propose a security layer placed between the routing and MAC layers, able to verify the freshness of a packet with different approaches, either based on time or on a unique packet identifier computed by combining time and the generating node address information with the HASH function (more details are presented later in this section). Regardless of the freshness mechanism employed, the security layer performs the two operations listed in the following and summarized in Figure 2.

- When generating a new packet, a 4 Bytes freshness index information is added to the packet (Figure 2(a)).
- When receiving a packet, the freshness of the received message is evaluated at the lower layer. If the packet passes the security check, it is forwarded to the upper layer, otherwise it is dropped (Figure 2(b)).

[2]Although in this paper the attacker cannot modify the content of the messages, it can still deflect messages by retransmitting broadcast packets.

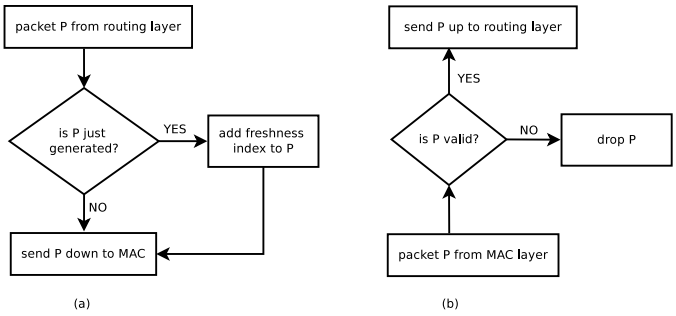

Fig. 2. Diagram describing the operations performed by the security layer: (a) packets arrived from the routing layer; (b) packet received from the MAC layer.

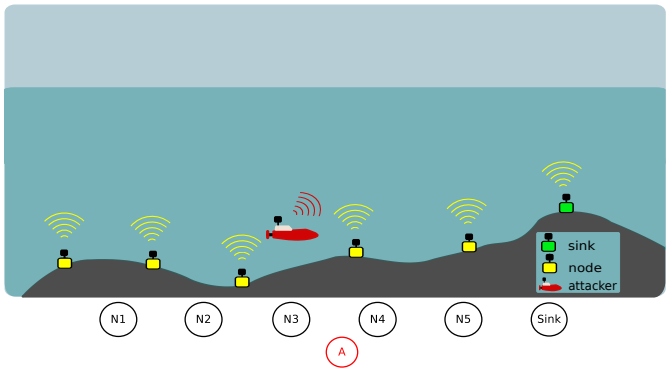

Fig. 3. Simulation scenario and topology of NET1. An AUV acts as an attacker and tries to saturate the underwater network in order to reduce the packet delivery ratio at the sink (green node).

Indeed, with the network protocols considered in this paper (flooding and static routing), when no security mechanisms are applied, whenever a node receives a packet from a neighbor, it forwards it to the next hop, without checking the content of the message. For this reason a security layer is required to check the freshness of a packet. The two replay attack countermeasures analyzed in this paper are named TIME and HASH, respectively.

1) TIME uses the packet generation time to verify the freshness of a packet: if the difference between the current time and the generation time of a received packet is above a pre-defined time threshold, the packet is discarded. This method requires the transmitter to store the packet generation time in an additional header with size 4 Bytes.
2) HASH computes the XOR operation between the HASH of the packet generation time and the HASH of the node address. This method requires the transmitter to store the HASH value in an additional header with size 4 Bytes: each time a packet is received by a node, the node checks in a HASH list whether a packet with the same HASH has already been received or not. If so, it discards the packet, otherwise the packet is forwarded to the upper layer and the value of the HASH is stored in the HASH list.

The HASH list has fixed size: once the list is full the HASH corresponding to the oldest packet is discarded: during the protocol evaluation the HASH list size required to ensure security to all replay attacks will be analyzed.

For the freshness information size, if we assume a maximum deployment time of one year, keeping the time precision in tenths of a second, we need at least 29 bits for the time representation, hence with 4 Bytes we can ensure the attacker needs to wait 10 years before the time index overflows. We also select 4 Bytes for the HASH index, not only because most of HASH operations return a 4 Bytes number, but also because it ensures a very low collision probability. For example, if the HASH list size is 2000 packets, the probability that at least two packets have the same hash value is less than 0.001.

III. Simulation Scenarios and System Settings

In this paper we consider two different scenarios, NET1 and NET2. NET1, illustrated in Figure 3, consists of a static linear network composed by 5 underwater nodes (depicted in yellow). The nodes are moored on the sea bottom with a distance between two consecutive nodes of 1 km. Each node generates and forwards packets to a common collection node (i.e., the sink, depicted in green). The AUV (depicted in red) plays the role of the malicious node, recording the packets received by the surrounding nodes and replaying them into the network. The impact of the attacker is analyzed when deployed in different positions in the network, and for both contention-free and contention-based MAC protocols, specifically Time Division Multiple Access (TDMA) (with time frame 6.5 s, equally divided between the 5 nodes) and Carrier Sense Multiple Access (CSMA). All the nodes in NET1 are equipped with a mediun frequency acoustic modem, with carrier frequency $f_c = 25$ kHz, bandwidth $BW = 5$ kHz, bitrate 4.8 kbps and a maximum range of 2.5 km. All nodes generate packets according to Poisson traffic, with packet size 125 Bytes and average inter-packet generation time 60 s. The analysis of this simple scenario is crucial to understand the impact of different versions of the replay attack in the network, as well as of the defense mechanism, varying the position of the attacker and attacking both contention-based and contention-free MAC layers. This first step will provide us with an idea on how to attack more complex networks, like the one depicted in Figure 4.

The second scenario (NET2), illustrated in Figure 4, consists of a hybrid network composed by 4 underwater nodes moored on the sea bottom, a ship and two AUVs (depicted in yellow). The distance between two adjacent moored nodes is 3 km, while AUV1 and AUV2 move at a constant speed of 1 m/s following a linear trajectory between node 2 and node 3, and between node 2 and 4, respectively. All nodes broadcast data to all other nodes using a flooding routing protocol: in this configuration, the maximum number of hops between the static nodes is 2, while with up to 3 hops all static nodes are able to reach the AUVs. Also in this case, the AUV depicted

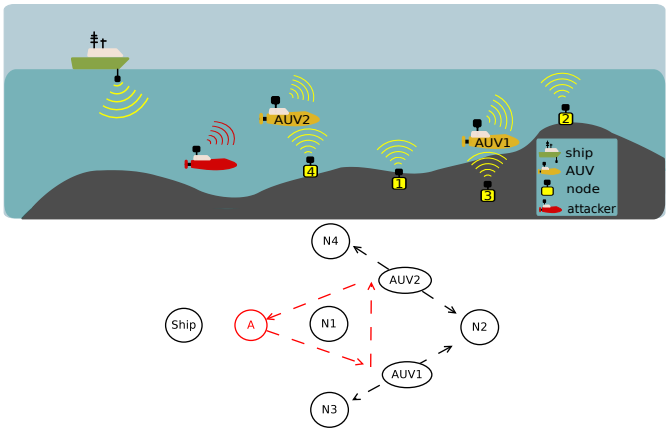

Fig. 4. Simulation scenario and topology of NET2. An AUV acts as an attacker and tries to saturate the underwater network in order to reduce the packet delivery ratio of the nodes.

TABLE I
SIMULATION SETTINGS

	NET1	**NET2**
Destination	unicast, single sink	broadcast
Nodes, hops	6, 5	7, 3
f_c, BW	25 kHz, 5 kHz	12 kHz, 10 kHz
Rate, range	4.8 kbps, 2.25 km	500 bps, 4.5 km
Traffic	each node Poisson: 125 Bytes, 60 s	7 types of traffic, total load: 565 bps
MAC	TDMA, CSMA	CSMA-ALOHA
Routing	static routing	flooding
Topology	linear, with consecutive nodes 1 km apart	hybrid with AUVs

in red plays the role of the malicious node, recording the packets received from the surrounding nodes and replaying them while moving around in the network, as shown in the topology depicted in Figure 4 (bottom part of the figure). All the nodes in NET2 are equipped with a low frequency acoustic modem, with carrier frequency $f_c = 12$ kHz, bandwidth $BW = 10$ kHz, bitrate 500 bps and a maximum range of 4.5 km.

In NET2, different traffic types are generated, all according to a Poisson traffic, specifically:

- AUV sensed data: each AUV generates two packets with size 60 Bytes each, on average, every 40 s to be broadcast to all other nodes;
- AUV status for the ship: each AUV generates two packets with size 60 Bytes each, on average, every 120 s to be sent in unicast to the ship;
- node status transmission: each static node (i.e., both moored nodes and the ship) generates a packet with size 32 Bytes, on average, every 120 s to be broadcast to all other nodes;
- ship position data: the ship generates one packet with size 60 Bytes, on average, every 90 s, to be broadcast to all other nodes;
- asynchronous messages: the ship generates one packet with size 60 Bytes, on average, every 120 s, to be sent in unicast to one of the AUVs;
- ranging information: each node generates one ranging packet with size 40 Bytes, on average, every 120 s, to be broadcast to all neighbors. This type of traffic is only transmitted to the one hop neighbors, without forwarding it to the next hops.

Both NET1 and NET2 have been evaluated using the DESERT Underwater Network simulator, and we used the channel model described in [23] with a spreading factor $k = 1.5$ (practical spreading), shipping activities equal to 0.5, no wind effects and a Binary Phase-Shift Keying modulation. The bit error rate (BER) is computed by applying the BER formula for the BPSK [24] to the signal to noise and interference ratio (SINR) computed by DESERT, and the packet error rate assumes independent and uniform distribution of the bit errors. Although more realistic channel models can be used inside the DESERT Underwater Framework [25], [26], we decided to employ this simple model to focus our analysis more on the effects of the replay attack itself than on the performance degradation experienced on a real acoustic channel due to multipath, Doppler and time varying noise.

IV. RESULTS

We now analyze the impact of the replay attack and the proposed countermeasures in networks NET1 and NET2.

A. Replay Attack and Countermeasures in NET1

For scenario NET1, we first analyze the impact of the replay attack when no security mechanisms are applied. Then, we evaluate the performance of the proposed countermeasures. With this scenario all nodes generate packets for the sink, with the specific intent of making the sink receiving all generated packets: for this reason we analyze the network performance in terms of Packet Delivery Ratio (PDR), i.e., the ratio between the number of packets received by the sink and the number of packets generated by all nodes of the network, without counting the duplicates.

1) Effect of the Replay Attack: The effect of the replay attacks in terms of PDR, when used in NET1 with TDMA at the MAC layer, is reported in Figure 5: these results are presented with 95% confidence intervals. In this case the TDMA frame is 6.5 s, equally divided between the 5 nodes. The replay node does not have a MAC layer and transmits a recorded packet every 2 s. Without the attacker, almost every packet is successfully received at the destination, thanks also to the fact that a TDMA MAC is contention free, and therefore avoids any interference. The MULTI-PACKET replay attack is more destructive when the attacker is positioned in the middle of the network. This happens because the attacker simply replays

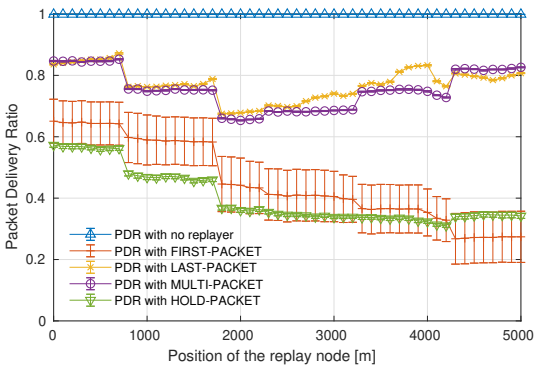

Fig. 5. Packed delivery ratio of the network versus the replay node position in NET1 with a TDMA MAC protocol. Position 0 m is close to the first node of the network, and position 5000 m is close to the network sink.

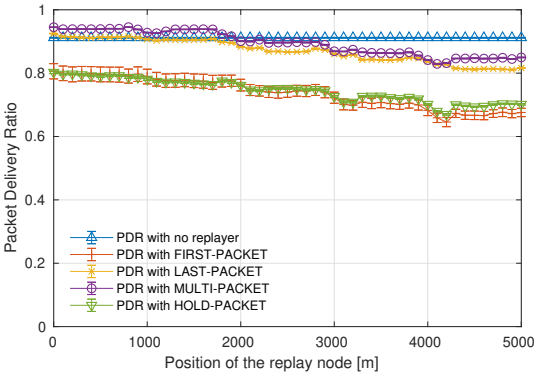

Fig. 6. Packed delivery ratio of the network versus the replay node position in NET1 with a CSMA MAC protocol. Position 0 m is close to the first node of the network, and position 5000 m is close to the network sink.

only once all the recorded packets. Observing the network from a central location allows the attacker to successfully record, and therefore transmit, more packets than when it observes the network from a peripheral location (e.g., close to the sink or to the first node of the network). In the FIRST-PACKET replay attack, the attacker always replays the first packet it recorded, by filling the MAC queue of the node that is required to forward that packet. The node forwards the attacker packet along the network, filling also the other nodes queue. This type of attack is more effective when the attacker is close to the sink, since the sink neighbors are the ones handling more packets. Therefore, the attacker can easily saturate the network exploiting this bottleneck. LAST-PACKET transmits the last packet sent in the network. For this reason, when the attacker with LAST-PACKET is near the sink, it sends, most of the times, packets received from the last relay, and therefore intended for the sink, which does not need to forward any packet. If the attacker is in a more centralized location, it affects the network performance similarly to the MULTI-PACKET attack. HOLD-PACKET, instead, fills uniformly the queues of all nodes in range of the attacker, as it selects at random which packet to retransmit. This is the most effective attack if the malicious node is deployed between the first and the fifth node of the network: then FIRST-PACKET becomes the most harmful, because during its packet selection HOLD-PACKET transmits also packets for the sink, that are therefore not forwarded by any of the nodes of the network.

Differently from the previous case, in NET1 with a CSMA protocol 9% of the packets are lost even without the presence of the attacker, due to interference caused by the contention-based MAC protocol. Despite this disadvantage, in this network there is a better use of the channel and, therefore, a bigger amount of traffic can be supported. The reason is due to the inefficiency introduced by the TDMA guard time, that is set equal to the propagation time experienced in the transmission between two adjacent nodes, and by the fact that in the first network a node can transmit only once within a time frame, and no parallel transmissions are scheduled, even if two nodes are separated by more than two hops. Therefore, with the same configuration analyzed before, the replay node does not overload the network as much as with the TDMA configuration. In addition, if the attacker is between the first and the third node of the network (i.e., when it is deployed between position 0 and position 2000 m) with LAST-PACKET and MULTI-PACKET it even improves the performance of the network, as it duplicates only packets that need to be transmitted for more hops, and, therefore, that have a higher probability of collision. Afterwords, it fills the packet queue of the nodes close to the sink, creating a bottleneck and thus causing a drop of 10% of the PDR. FIRST-PACKET and HOLD-PACKET, instead, always provide a drop of the PDR, even when the attacker is close to the first node, as they replicate old packets that have already been processed many hours before, therefore their retransmission does no provide any benefits to the network. Also in this case the attacker is more effective when placed close to the last relay of the network.

2) Replay Attack Countermeasures: In this subsection we focus on the replay attack contermeasures for NET1 configured with a TDMA MAC layer. Figure 7a provides the PDR of the network under the FIRST-PACKET attack when changing the malicious node position without countermeasure (blue line), and with TIME (red line with X marker) and HASH (green line with round marker) defense mechanisms. With this attack, both countermeasures provide similar results, i.e., a 30% increase of the PDR compared to the case without defense mechanisms. A similar result is obtained for the HOLD-PACKET attack (Figure 7d), where the increase in PDR is 40% compared to the case without defense mechanisms. HOLD-PACKETS stores in a buffer the first 20 packets it receives, and after 3600 s it starts transmitting one of them at a time, selected at random, for the whole simulation. With this attack, the performance of TIME and HASH configured with a HASH list of 30 HASH values is very similar. In the case of a HASH list with size less than 20, instead, TIME outperforms HASH (Figure 8, yellow line), because if a very old packet is transmitted it will not be present in the HASH list anymore, while TIME is unaffected by this issue. On the other hand, the

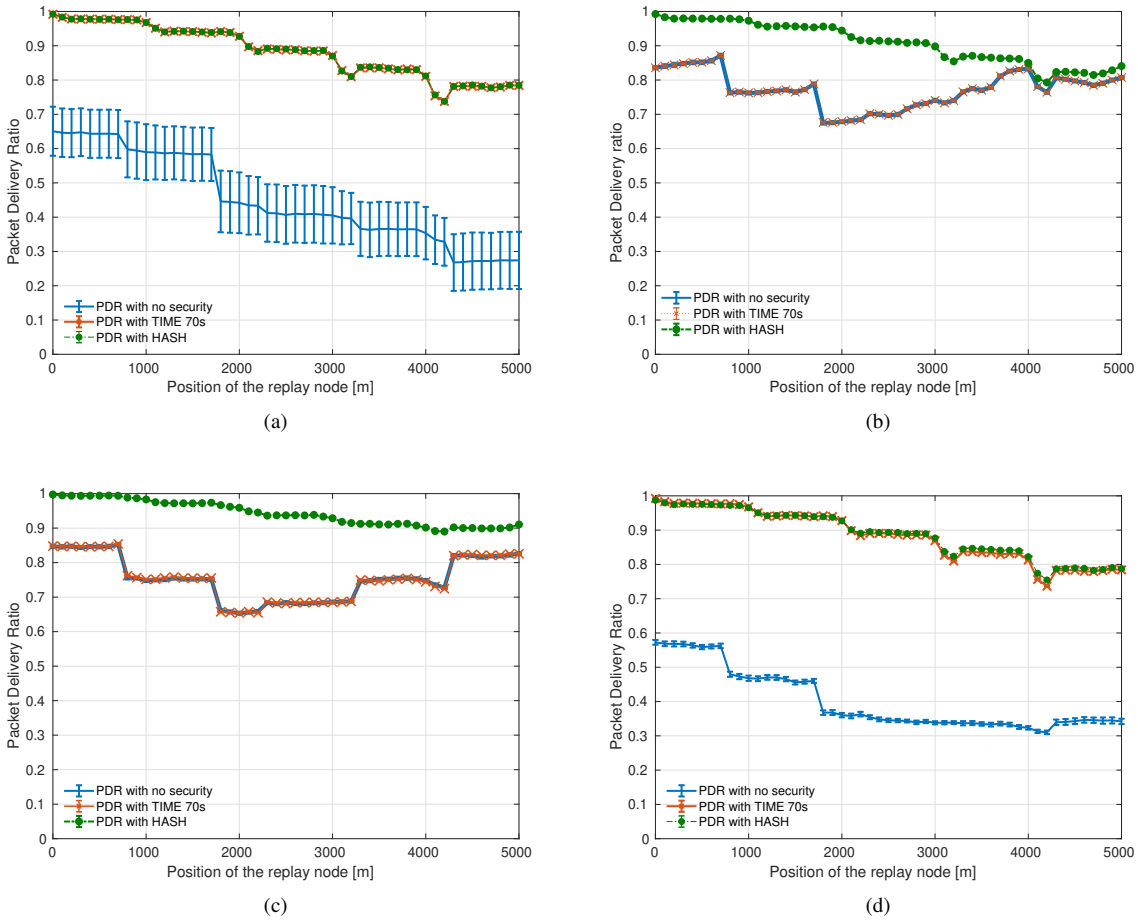

Fig. 7. Packed delivery ratio of the network versus the replay node position in NET1 with a TDMA MAC protocol and different countermeasures. (a): FIRST-PACKET; (b) LAST-PACKET; (c): MULTI-PACKET; (d) HOLD-PACKET.

TIME countermeasure is substantially ineffective for LAST-PACKET and MULTI-PACKET attacks (depicted in Figure 7b and Figure 7c, respectively). Indeed, this countermeasure, with 70 s of packet validity time, is not able to drop the repetitions sent by the replay node. However setting the validity period to a lower value would lead the countermeasure to drop also legitimate packets, leading to even worse results: the value 70 s has been chosen as the minimum value of packet validity time to ensure no legitimate packets are discarded in NET1. HASH, instead, is immune to this phenomenon, however its PDR suddenly decreases when the distance between attacker and sink decreases, as in this case the malicious node is attacking the network in an area that is closer to its bottleneck, i.e., the last node before the sink. Indeed, this node has more traffic to deliver to the destination. In this case the PDR decreases from 99% when the attacker is between the first and the second node, down to 80% when the attacker is placed close to the sink of the network, due to packet collisions caused by a collateral jamming effect. This effect is attenuated in the MULTI-PACKET attack (Figure 7c), as the packet transmissions are limited to the number of packets received by the attacker.

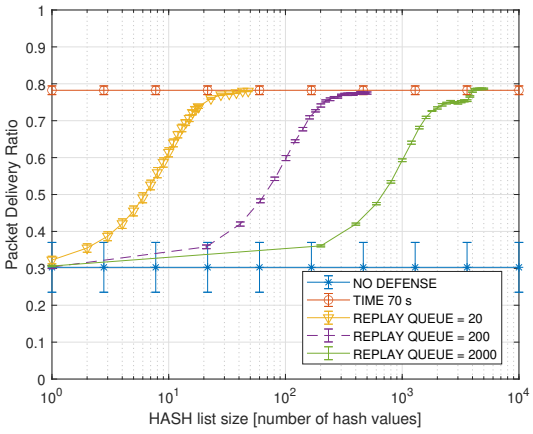

Fig. 8. PDR of NET1 under HOLD-PACKET attack with different configurations of the attacker queue size and the HASH list stored in the nodes. The attacker is deployed between the sink and the last node before the sink.

To investigate more in depth when HASH outperforms TIME as countermeasure of the HOLD-PACKET replay at-

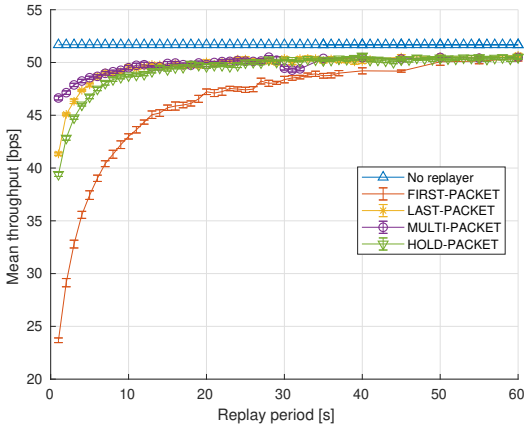

Fig. 9. Average throughput received by one node versus the replay period in NET2.

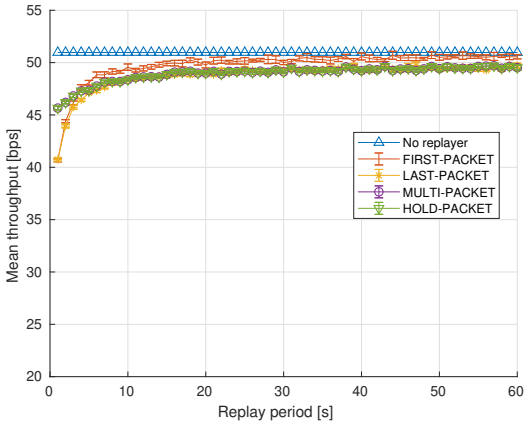

Fig. 10. Average throughput received by one node versus the replay period in NET2 with TIME countermeasure.

tack, we test different configurations of the attacker packet queue size and the HASH list size when the attacker is deployed between the sink and the last node before the sink, i.e., in the most advantageous configuration for the malicious node.

Figure 8 demonstrates that, as soon as the number of hashes stored in the HASH list is greater than or equal to 2 times the number of packets recorded by the attacker, the performance of HASH and TIME are equal. We remark that while the attacker needs to store the whole packet, the defender requires to store only the HASH value, that is 4 Bytes per packet. For instance, if the attacker has a queue of 2000 packets, the nodes are required to allocate only 16 kBytes to store 4000 hash values. Once this requirement is satisfied, in NET1 HASH is never outperformed by TIME, for all attack strategies.

From our analysis, the same network with a CSMA MAC layer provides similar results, but is less interesting to analyze because such network configuration is less affected by the attack.

B. Replay Attack and Countermeasures in NET2

Also for scenario NET2 we first analyze the impact of the replay attack when no security mechanisms are applied. Then, we evaluate the performance of the proposed countermeasures. With this scenario all nodes generate packets for all other nodes, with the intent of making the other nodes receive fresh position and data updates. In this scenario the data is carrying status updates and repeated over time, thus if some packets are lost a newer status with updated information can be obtained from subsequent messages. In this case we analyze the network performance in terms of average throughput per node, defined as the average number of payload bits that arrive to a node in a second, without counting the duplicates.

1) Effect of the Replay Attack: The effect of the replay attacks in terms of average throughput received by one node when used in NET2 is reported in Figure 9: these results are presented with 95% confidence intervals. The replay node does not have a MAC layer, and the effect of its attack is analyzed by varying the replay period, from 1 s to 60 s. Without the attacker, the throughput is 51.68 bps.

The MULTI-PACKET replay attack is the least destructive because it replays only once all the recorded packets, i.e., the maximum number of packets it transmits is bounded by the number of packets it receives. In addition, if the retransmitted packet was not received by one of the nodes in range due to packet collisions, when this packet is received by the application layer it is not discarded.

FIRST-PACKET, on the other hand, is the most destructive of the replay attacks considered in this paper when no countermeasure is applied. This happens because this attack always injects the same old packet into the network, therefore increasing the network traffic without adding any packets that might have been lost by other nodes. In HOLD-PACKET, instead, after 6000 s of recordings where the attacker saves up to 2000 of the received packet in a buffer, it transmits packets randomly chosen from the 2000 packets stored: similarly to MULTI-PACKET, if the transmitted packet was not received by a node, it is not discarded by its application layer. HOLD-PACKET with a buffer size of 1 packet will behave instead like FIRST-PACKET: although at this point having a small buffer seems like the best choice, in the next section we will see that it is not, as the countermeasure of FIRST-PACKET is very easy.

Finally, with LAST-PACKET the attacker always replays the last packet it receives, almost acting as a relay of the flooding network. The node forwards the attacker packet along the network, filling also the other nodes queue.

2) Replay Attack Countermeasures: Figures 10 and 11 demonstrate that TIME and HASH countermeasures are effective also in NET2, with flooding routing and mobile nodes. In this case, with TIME countermeasure the time validity of a packet is set to 150 s. TIME and HASH countermeasures perform similarly against FIRST-PACKET, providing a throughput increase up to 17 bps (70% throughput increase compared to the throughput without countermeasures) when

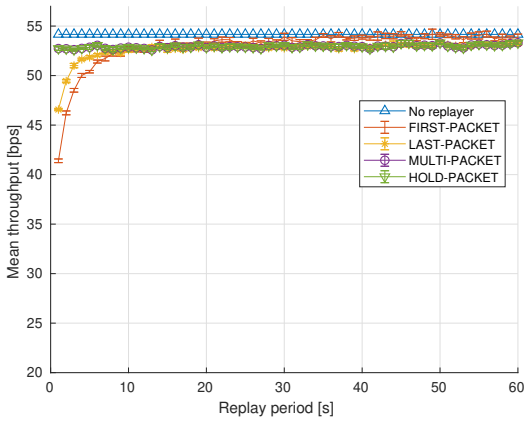

Fig. 11. Average throughput received by one node versus the replay period in NET2 with HASH countermeasure.

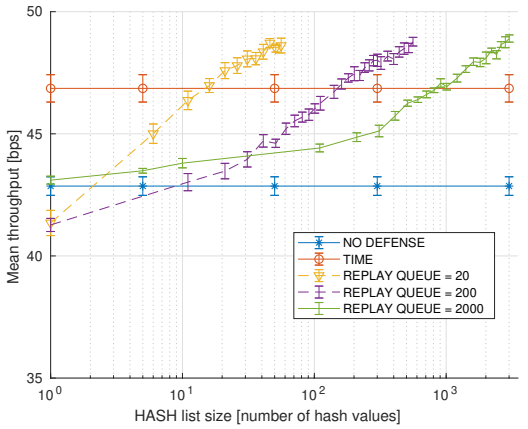

Fig. 12. Throughput of NET2 under HOLD-PACKET attack with different configurations of the attacker queue size and the HASH list stored in the nodes.

the replay period is less than 5 s. Increasing the replay period, the benefits of the network with countermeasures compared to the network without countermeasures decrease, and so does the effect of the attack. A similar behavior is observed against the HOLD-PACKET attack, where both countermeasures provide a throughput increase of 5 bps (12.5% throughput increase compared to the throughput without countermeasures), when the replay period is less than 5 s. In this network, TIME does not provide benefits against LAST-PACKET and MULTI-PACKET, while for short replay period HASH provides a throughput increase of 5 bps (12%) against LAST-PACKET and of 8 bps (17.7%) against MULTI-PACKET.

To investigate when HASH outperforms TIME as countermeasure of the HOLD-PACKET replay attack, we test different configurations of the attacker packet queue size and the HASH list size also in the case of NET2, when the attacker uses a packet transmission period of 3 s. Figure 12 demonstrates that, as soon as the number of hashes stored in the HASH list is greater than or equal to the number of packets recorded by the attacker, HASH outperforms TIME. Once this requirement is satisfied, in NET2 HASH is never outperformed by TIME, no matter the attack strategy.

V. Conclusion and Future Work

In this paper, we studied the effect of four different versions of the classical replay attack in UANs. We proposed two countermeasures, TIME and HASH. The former was based on the packet generation timestamp and the latter was based on the HASH value of the packet generation timestamp combined with the address of the source node. While the TIME solution is very effective if the attacker replays only old packets, generated several minutes (depending on the maximum packet delivery delay of the network) before the reception, the HASH countermeasure limits a lot the effect of the attack also in the case the attacker replays very recent packets. We demonstrated that the proposed countermeasures perform efficiently in a linear network with a unique sink and an static attacker as well as in a broadcast network with mobile/static nodes and a mobile attacker. Future work will focus on evaluating the proposed system through a field test evaluation.

References

[1] 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 2007.

[2] 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.

[3] F. Chiariotti, C. Pielli, N. Laurenti, A. Zanella, and M. Zorzi, "A game-theoretic analysis of energy-depleting jamming attacks with a learning counterstrategy," *ACM Transactions on Sensor Networks (TOSN)*, vol. 16, no. 1, pp. 1–25, Nov. 2019.

[4] E. C. H. 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, 2006, pp. 3383–3389.

[5] 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.

[6] 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.

[7] A. Signori, C. Pielli, F. Chiariotti, F. Campagnaro, M. Giordani, N. Laurenti, and M. Zorzi, "Jamming the underwater: a game-theoretic analysis of energy-depleting jamming attacks," in *ACM International Conference on Underwater Networks & Systems (WuwNet)*, Oct. 2019.

[8] 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.

[9] 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.

[10] G. Dini and A. L. Duca, "A secure communication suite for underwater acoustic sensor networks," *Sensors*, vol. 12, no. 11, pp. 15 133–15 158, November 2012.

[11] A. Caiti, V. Calabro, G. Dini, A. Lo Duca, and A. Munafo, "Secure cooperation of autonomous mobile sensors using an underwater acoustic network," *Sensors*, vol. 12, no. 2, pp. 1967–1989, February 2012.

[12] Y. Liu, J. Jing, and J. Yang, "Secure underwater acoustic communication based on a robust key generation scheme," in *Proceedings of the 9th International Conference on Signal Processing*, ser. ICSP'08, Leipzig, Germany, 10–11 May 2008, pp. 1838–1841.

[13] G. Ateniese, A. Capossele, P. Gjanci, C. Petrioli, and D. Spaccini, "SecFUN: Security Framework for Underwater acoustic sensor Networks," in *Proceedings of MTS/IEEE OCEANS 2015*, Genova, Italy, 18-21 May 2015, pp. 1–9.

[14] F. Campagnaro *et al.*, "The DESERT underwater framework v2: Improved capabilities and extension tools," in *Proc. UComms*, Lerici, Italy, Sep. 2016.

[15] P. Syverson, "A taxonomy of replay attacks [cryptographic protocols]," in *Proc. The Computer Security Foundations Workshop VII*, Franconia, NH, USA, Jun. 1994, pp. 187–191.

[16] K. Murakami, H. Suemitsu, and T. Matsuo, "Classification of repeated replay-attacks and its detection monitor," in *IEEE 6th Global Conference on Consumer Electronics (GCCE)*, 2017, pp. 1–2.

[17] Y. Mo and B. Sinopoli, "Secure control against replay attacks," in *47th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2009, pp. 911–918.

[18] S. Malladi, J. Alves-Foss, and R. Heckendorn, "On preventing replay attacks on security protocols," *Proc. International Conference on Security and Management*, 06 2002.

[19] D. E. Denning and G. M. Sacco, "Timestamps in key distribution protocols," *Commun. ACM*, vol. 24, no. 8, p. 533–536, Aug. 1981. [Online]. Available: https://doi.org/10.1145/358722.358740

[20] F. Farha and H. Ning, "Enhanced timestamp scheme for mitigating replay attacks in secure zigbee networks," in *IEEE International Conference on Smart Internet of Things (SmartIoT)*, 2019, pp. 469–473.

[21] F. Farha, H. Ning, S. Yang, J. Xu, W. Zhang, and K. R. Choo, "Timestamp scheme to mitigate replay attacks in secure zigbee networks," *IEEE Transactions on Mobile Computing*, Jul. 2020, Early Access.

[22] D. Jinwala, D. Patel, S. Patel, and K. S. Dasgupta, "Replay protection at the link layer security in wireless sensor networks," in *WRI World Congress on Computer Science and Information Engineering*, vol. 1, March 2009, pp. 160–165.

[23] 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.

[24] N. Benvenuto and M. Zorzi, *Principles of Communications Networks and Systems*, 1st ed. Wiley, 2011.

[25] "The World Ocean Simulation System - *WOSS*," Last time accessed: Aug 2020. [Online]. Available: http://telecom.dei.unipd.it/ns/woss/

[26] P. Casari, F. Campagnaro, E. Dubrovinskaya, R. Francescon, A. Dagan, S. Dahan, M. Zorzi, and R. Diamant, "ASUNA: A topology dataset for underwater network emulation," *IEEE Journal of Oceanic Engineering*, March 2020, Early Access.