Auth-AIS

Citation for published version (APA):

Document license:
TAVERNE

DOI:
10.1109/TDSC.2021.3069428

Document status and date:
Published: 01/07/2022

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher’s website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 16. Sep. 2023
Auth-AIS: Secure, Flexible, and Backward-Compatible Authentication of Vessels AIS Broadcasts

Savio Sciancalepore®, Pietro Tedeschi®, Ahmed Aziz, and Roberto Di Pietro®

Abstract—Automatic Identification System (AIS) is the de-facto communication standard used by vessels to broadcast identification and position information. However, being AIS communications neither encrypted nor authenticated, they can be eavesdropped and spoofed by adversaries, leading to potentially threatening scenarios. Existing solutions, including the ones conceived in the avionics domain, do not consider integration with the AIS standard, and they do not provide protection against rogue messages flooding. In this article, we propose Auth-AIS, a secure, flexible, standard-compliant, and backward-compatible authentication framework to secure AIS broadcast messages. Auth-AIS leverages existing sound cryptographic tools, including TESLA and Bloom Filters, inheriting their security properties while contextualizing them in the AIS technology. Auth-AIS is a software-only solution, that can be seamlessly integrated into existing AIS deployments, without requiring any hardware replacement. Its innovative design also provides backward-compatibility—i.e., Auth-AIS messages can be received also by AIS users not adopting Auth-AIS, while renouncing at its security guarantees. Auth-AIS can work in either two configuration modes: Deterministic Security Configuration, able to achieve low-delay authentication with a message overhead of 75 percent, or Probabilistic Security Configuration, reducing the message overhead down to 35.71 percent, while experiencing a marginal increase in the authentication delay. All these security configurations guarantee an 80 bits equivalent security level and false-positive rate less than 2−30. Note that these latter security parameters can easily be tuned to fit different security requirements. Finally, the source code of Auth-AIS in the GNU Radio ecosystem has been released as open-source, to foster research activities from both Industry and Academia on secure AIS communications.

Index Terms—AIS, vessels cybersecurity, broadcast authentication, cyber-physical systems security

1 INTRODUCTION

The Automatic Identification System (AIS) is the communication standard used nowadays by vessels to broadcast information about their position, velocity, and identification, as well as to manage self-separation and exchange short audio messages [1]. Since 2002, the integration of AIS transceivers is mandatory on all commercial vessels exceeding 300 tonnes, and nowadays over 1,644,000 vessels use AIS for regular operations [2].

Despite its increasing usage and importance in navigation decisions [3], AIS does not provide any security feature [4]. This is because AIS was conceived in the early 1990s, with availability as the only requirement. Indeed, AIS messages are transmitted in clear-text and they are not authenticated. Therefore, any malicious adversary equipped with either a compatible AIS transceiver or an ad-hoc Software Defined Radio (SDR) can easily send forged messages pretending to be a legitimate vessel. The above vulnerability can lead to threatening scenarios, such as forcing vessels to change their route [5]. In turn, this can lead to further threatening scenarios, such as preventing ships to rescue distressed vessels.

A few contributions in the literature already identified the security weaknesses of AIS, and proposed some countermeasures. As detailed in Section 2.2, regarding security strategies working directly on the vessel, the proposed solutions introduce security features in the AIS communication technology without considering the related system requirements. In particular, these contributions either modify the existing AIS standard, or they propose solutions that require an overwhelming message overhead, resulting in high deployment costs and low suitability for congested scenarios, such as ports. Despite potential solutions have been conceived in the avionic context for the Automatic Dependent Surveillance - Broadcast (ADS-B) communication technology, they cannot be applied to AIS directly, and they do not support resilience to message flooding attacks (see Section 2.2 for more details).

Contribution. To overcome the previously-cited gaps, in this paper, we propose Auth-AIS, a backward-compatible authentication scheme for vessels AIS broadcasts. Auth-AIS can be seamlessly integrated into existing AIS deployments as a simple software update, not requiring any hardware modification or service interruption to AIS transceivers already mounted on-board of commercial vessels. Auth-AIS is a flexible security framework, that can tune the provided security based on the reporting rate of the vessel and
congestion level of the underlying communication channel. When the channel is almost free, Auth-AIS can be configured using the Deterministic Security Configuration, which is based on the integration of the well-known TESLA symmetric authentication protocol in the AIS technology. When the channel shows an increased level of congestion, Auth-AIS can be configured using the Probabilistic Security Configuration, based on the coupling of TESLA with the Bloom Filter tool. Overall, Auth-AIS emerges as a very bandwidth-efficient and secure solution, being able to guarantee a minimum security level equivalent to 80 bits (according to the definition of security level reported by NIST [6]) and false-positives rates less than $2^{-40}$, while experiencing a message overhead of 35.71 percent. When a trade-off between security, bandwidth overhead, and authentication delay is required, and security is at premium, a security level of 192 bits can be guaranteed with a message overhead of 40 percent—these results are drawn from an extensive experimental campaign. Comparisons against competing solutions show the viability and efficiency of our solution. Auth-AIS has been fully implemented in the GNU Radio ecosystem, and the source code has been released as open-source [7]. This enables the research community to verify our claims, and use the released code as a ready-to-use basis for further protocol improvement and comparison.

We highlight that our work's novelty is in the provisioning of authentication security services to the AIS communication technology, in a way to be standard-compliant, backward-compatible, and flexible to the several operation conditions of vessels (ports and off-shore). We also show an experimental proof-of-concept, demonstrating that the software of current AIS transceivers could be updated to support our security solution, while at the same time enjoying backward compatibility with the devices that—for any reason—have not been updated. Moreover, although Auth-AIS integrates well-known tools such as TESLA and Bloom Filters, we innovatively combine them. Indeed, the provided solution that allows achieving delayed authentication in low-bandwidth communication technology has not been adopted before in the literature (see Section 2.2).

Roadmap. The rest of this paper is organized as follows. First, Section 2 introduces AIS, the tools used in our paper, and reviews the related work. Section 3 provides some crucial system requirements to be fulfilled by any security solution for AIS. Then, Section 4 depicts the scenarios and the adversary model assumed in this work, Section 5 provides the logic and the details of Auth-AIS, while Section 6 analyzes the security of the provided solution. The performance evaluation of Auth-AIS is discussed in Section 7, while Section 8 tightens the conclusions.

## 2 Preliminaries and Technical Background

In this section, we introduce preliminary concepts, useful to catch the logic of our proposal. Specifically, Section 2.1 introduces background details on the AIS communication technology, Section 2.2 reviews some related work on security issues for AIS, Section 2.3 introduces the TESLA authentication protocol and, Section 2.4 reviews the most important concepts driving the adoption of Bloom Filters.

### 2.1 Automatic Identification System

The AIS communication protocol has been developed in the early 1990s, to allow vessels and shore stations to exchange navigation information. Since the year 2002, the SOLAS Agreement enacted by the International Maritime Organization (IMO) mandated the integration of a Class-A AIS transceivers on-board of all vessels exceeding a gross tonnage of 300 tonnes [5].

Nowadays, vessels use AIS for many services. First, vessels use AIS regularly to broadcast identification and position data, acquired via regular Global Navigation Satellite System (GNSS) technologies (e.g., GPS). This feature enables several applications, such as identification, route adjustment, accident prevention and investigation, search and rescue operations, and remote tracking. Besides, two vessels can establish a one-to-one communication, exchanging dedicated binary messages. We notice that some AIS transceivers are also mounted within ports and along the coastal tracks, to enhance ship tracking and assist in search-and-rescue operations.

From the physical-layer perspective, AIS communications use the Very High Frequency (VHF) frequency band, and specifically, the 161.975 MHz (Channel A) and 162.025 MHz (Channel B). The modulation scheme is the Gaussian Filtered Minimum Shift Keying (GMSK), with a bit rate of 9,600 bits/sec. The theoretical maximum transmission range that can be reached by commercial AIS transceivers is approximately 70 km, even if weather factors can reduce it to 40 km. To overcome this issue, recently AIS transceivers have been installed on general-purpose Low-Earth Orbit (LEO) satellites, leading to the Space-based AIS (SAT-AIS) communication paradigm, and extending the transmission range up to 400 km [8], [9].

The standard format of AIS messages is depicted in Fig. 1.

AIS messages have an overall size of 256 bits [10]. The message starts with the first 8 bits reserved to turn on the AIS transceiver. Then, the message contains a preamble of 24 bits, used to identify an incoming AIS message, and for the receiver to synchronize with the symbols emitted by the transmitter. A Start Flag of 8 bits denotes the start of the AIS data message, having a regular size of 168 bits only. The bottom part of Fig. 1 shows the content of AIS binary broadcast messages (Message Type 8), being these messages the ones specifically considered in this work. They contain the Message Type (6 bits), indicating the type of the message, the Repeat Indicator (2 bits), used by the AIS repeater(s) to indicate how many times a message has been retransmitted, the Source ID Maritime Mobile Service Identity (MMSI) (30 bits), uniquely identifying the transceiver emitting the
message, a Spare field (2 bits) reserved for future use, the Designated Area Code of 10 bits, which is a jurisdiction code (it uses the same encoding as the area designator in MMSIs), the Functional ID of 6 bits, specifying the message sub-type, according to the IALA Recommendation e-NAV 144, the Application Data of 80 bits, dedicated to binary data, and finally, the last 32 bits devoted to bit stuffing. The Frame Check Sequence (FCS) field follows the AIS data message and it is used to provide error detection, via a Cyclic Redundancy Check (CRC) polynomial of 16 bits. The End Flag byte closes the transmission of the message. Finally, the Buffer field of 24 bits should be adopted only for bit-stuffing, distance delay and synchronization jitter. Overall, the time from the transmission of the first bit of the preamble to the end flag is always less than 26.667 ms (duration of the AIS time-slot).

Note that AIS RF operations take place according to a slot-based Time Division Multiple Access (TDMA) schedule. Specifically, the time is divided into slots, lasting 26.667 ms (coincidental with the duration of a packet). Each AIS transceiver periodically acquires the knowledge of the AIS entities in the radio neighborhood, by passively listening to one of the two AIS channels for 1 minute (2,250 slots). Based on this knowledge, it selects for transmission one or more free slots, i.e., time-slots where no RF activities were detected. According to the amount of data to be transmitted, one or more time-slots can be required. The standard mandates that a transceiver can continuously transmit data for a maximum number of consecutive slots, as a continuous data stream, depending on its class. Class A devices can transmit continuously for 5 consecutive slots, while transmitting Class B devices can occupy the medium for a maximum of 3 consecutive slots. The overall number of binary data bytes that can be transmitted continuously are summarized in Table 1. We refer the interested readers to the AIS standard specification for more details about the channel logic of AIS [10, p. 116].

2.2 Related Work

Several papers in the last years already identified the security weaknesses characterizing the AIS technology [5], [11], [12], [13]. At the same time, a few contributions proposed innovative security schemes to overcome such weaknesses. For instance, the authors in [14] proposed a key agreement scheme to secure pairwise vessels communications performed through the AIS communication technology, based on the combined usage of implicit certificates and Diffie-Hellman (DH) schemes. The authors in [15] introduced a three-layers pseudonym-based authentication scheme, providing source authenticity and anonymity at the same time. Similarly, the authors in [16] proposed a hash-based authentication strategy, while the authors in [17] leveraged digital certificates to provide source authentication and trajectory anonymization based on k-anonymity. Finally, the authors in [18] discussed the integration of Identity-Based Encryption (IBE) schemes to authenticate and encrypt vessel broadcasts, while the authors in [19] introduced new AIS message types to convey a digital signature of each AIS clear-text message emitted by a vessel.

However, we notice that the proposals in [15], [16], [17], and [18] did not discuss how to integrate the proposed security scheme in the existing AIS deployments. Indeed, they focused on the logic and the security properties of the proposed strategies, without contextualizing them in the current format of AIS packets. Therefore, they did not evaluate explicitly the message overhead of their solutions, as well as the consequences of their integration on existing deployments. For instance, solutions such as the ones proposed in [15], [16], [17], and [18] would require expensive and time-consuming modifications of all the existing AIS transceivers, preventing any legacy receiver to take part in the network. The only work discussing integration into the current AIS message format is [19]. However, integrating such a scheme into AIS would require an additional AIS message for each packet emitted by the vessel, with a resulting overhead of 50 percent. Moreover, this overhead is fixed, independently of the operating conditions of the vessel.

Therefore, at the time of this writing, we found that none of the solutions in the current literature have proposed security frameworks for AIS communications that can be integrated as simple software updates, without requiring hardware modifications to existing deployments. Besides, none of the actual proposals considered the high dynamicity of AIS communications, characterized by very different reporting rates and transmitters density, based on the underlying communication scenario.

The rationale of AIS is very similar to the ADS-B communication technology, used by commercial and military aircraft to broadcast identification and position information. Similarly to AIS, ADS-B-equipped aircraft emit unauthenticated and unencrypted broadcast packets, that could be spoofed by an adversary equipped with cheap SDRs. Therefore, several contributions in the last few years proposed solutions to address the cited security issues. Some contributions proposed physical-layer position verification solutions, aimed at cross-checking the information contained in ADS-B packets with physical-layer side information, such as the Doppler Effect [20], the Time-of-Arrival (ToA) [21], and the raw IQ signals [22]. Despite not requiring any modification on AIS transceivers, these solutions usually require additional dedicated hardware at the receiver side.

Conversely, other solutions integrated authentication techniques based on cryptography schemes in ADS-B packets. Specifically, the potential of TESLA to solve the ADS-B broadcast authentication issues has been highlighted by the authors in [23], and some contributions, such as [24] and [25], also provided prototype implementations. However, these approaches are not standard-compliant. Recently, the authors in [26] proposed SOS, a backward-compatible and standard-compliant protocol that provides...
authentication for ADS-B messages using TESLA, while not requiring any modification to the existing hardware, but only a software update. However, we notice that SOS is highly vulnerable to Denial of Service (DoS) attacks, where the adversary injects a large number of forged packets on the wireless communication channel. In Section 7.2 we will provide a detailed performance comparison between Auth-AIS and the SOS solution, showing the improvements of Auth-AIS in terms of authentication rate.

Finally, we would like to highlight that some contributions in the literature already integrated TESLA and the Bloom Filter tool in other application scenarios. For instance, the authors in [27] adopt compressed Bloom Filters to combine multiple μTESLA instances. Besides, the authors in [28] integrated a modified version of the Bloom Filter and the TESLA protocol, to protect Wireless Sensor Networks (WSNs) from Denial of Service attacks, at the expense of a small false negative rate introduced in the construction of the Bloom Filter. Similarly, the authors in [29] performed key aggregation for TESLA and managed collisions by using an XOR-Based Bloom Filter data structure. Indeed, multiple hash digests are first XORed and then used in a Bloom Filter as search indexes. Overall, security issues similar to the ones discussed in this paper have been tackled also in the context of Vehicular and Bluetooth-Low-Energy networks, e.g., by the authors in [30] and [31]. However, these communication technologies are very different from AIS, since AIS is evidently more low-rate and bandwidth-limited (9.6 Kbps). Therefore, despite the security problems do share some similarities, the unique features of AIS scenarios make this field challenging and interesting on its own.

### 2.3 Timed Efficient Stream Loss-tolerant Authentication

Timed Efficient Stream Loss-tolerant Authentication (TESLA) is a broadcast authentication protocol initially conceived by Perrig in [32]. The purpose of TESLA is to provide an efficient source authentication protocol for broadcast communications in very resource-limited environments, by leveraging only symmetric cryptographic primitives. Overall, TESLA requires the following assumptions at the system level: (i) a loose time synchronization between all the entities in the network, (ii) the division of the time into slots, and (iii) the deployment of a Trusted Third Party (TTP), in charge of assigning, renewing, and managing symmetric cryptographic keys assigned uniquely to each device in the network.

At the boot-up of the network, any entity receives from the TTP a master key, namely $K_M$, and a value $l$, that is a very large number, denoting the length of a cryptographic hash chain. These values are assumed to be trusted, and they are stored locally on the generic network element. Based on the above values, each device generates a cryptographic key chain, by recursively applying a hashing function $H$. Specifically, the key of a generic time-slot $i$ is generated according to Eq. (1):

$$K_i = H^{n+i}(K_M),$$

where $n$ is a large integer value defining the length of the cryptographic hash chain, while $H^n(\cdot)$ refers to the recursive execution of the hashing function $H$ for $n$ times.

The initial key $K_0$, namely the root key, is obtained according to Eq. (2):

$$K_0 = H^n(K_M).$$

Note that the root key is published by the transmitter, and it is publicly available to allow for key chain verification.

Then, each key $K_i$ is meant to be used in a dedicated time-slot $t_i$, to authenticate all the messages delivered by the network element in that time-slot. Specifically, in a generic time-slot $t_i$, a device transmits a given message $m$ and a message digest $h_i$, computed as in Eq. (3):

$$h_i = H(m, K_i),$$

where $K_i$ is the key computed according to Eq. (1). Note that, until the expiration of the time-slot $t_i$, only the legitimate transmitter is assumed to know the time-slot key $K_i$. Therefore, all the messages received from that network entity are temporarily stored.

Let us define the parameter $l$, namely the disclosure lag. It is the number of time-slots that the transmitter waits before unveiling the key $K_i$ used in the generic time-slot $i$ to authenticate all the messages transmitted in that time-slot. Therefore, after a given disclosure lag of $l$ time-slots, the transmitter unveils the key of the time-slot $t_i$, namely $K_i$, by transmitting it in clear-text. Only at that time, any receiver could verify the authenticity of the messages transmitted in the time-slot $t_i$, by checking that Eq. (3) is verified. At the same time, any receiver can verify the authenticity of the key $K_i$, by verifying the following Eq. (4):

$$K_0 == H^l(K_i).$$

Note that the digest and the key of any specific time-slot $t_i$ are delivered every time at the time-slot $t_{i+l}$. Therefore, even if the authentication information of one slot are lost due to collisions or interferences, the following time-slots are not affected.

### 2.4 Bloom Filters

A Bloom Filter is a space and time efficient probabilistic data structure, originally conceived to perform membership queries on large data sets [33]. Bloom Filters support two operations: the element mapping, dedicated to the insertion of elements inside the Bloom Filter, and the membership check, aimed at checking whether a given element may or is definitively not in the set [34]. More in detail, a Bloom Filter is a bit-array vector $BF$, composed of $m$ bits initially all set to 0.

Let us define a data-set $E = \{e_1, e_2, \ldots, e_J\}$, made up of $J$ elements. Besides, we define a set $\Omega = \{H_1, H_2, \ldots, H_k\}$ of $k$ independent hash functions, with a range from 0 to $m - 1$, and adopted to hash the values defined in the set $E$.

The element mapping operation is devoted to the insertion of an element into the Bloom Filter, and it requires the following two steps. First, each element $e \in E$ is hashed leveraging $k$ hashing functions. Next, selecting $1 \leq i \leq k$ and $1 \leq j \leq J$, all the bits at the Bloom Filter location $BF[H_i(e_j)]$ are set to 1. In the rest of this paper, assuming a generic input element $e_j$, we denote this operation as in the following Eq. (5):
\[ b = BF(e_j). \] (5)

The membership query operation is committed to check the presence of an element into the Bloom Filter. To query an element \( e \) in the Bloom Filter, a checking procedure to the bits at positions \( H_i(e) \) is executed as follows: if any of the bits at these positions is equal to 0, the element is definitively not in the set; otherwise, if all the bits in these positions are set to 1, then the element may be in the set.

We notice that the membership check operation cannot have false negatives, i.e., if an element is in the set, the membership check operation cannot return a 0. Conversely, the membership check operation could have false positives due to the collisions about the outputs of the hashing functions, i.e., there is a probability that even if an element is not in the set, the operation returns a 1, indicating that the element is in the set. The false positives rate, namely \( \psi \), can be mitigated by finely tuning the parameters of the Bloom Filter, i.e., the number of elements \( n \), the size of the Bloom Filter \( m \), and the number of hashing functions \( k \).

As reported by [33], the relationship between all the above-reported elements is aligned to the following Eq. (6).

\[ \psi = (1 - e^{-\frac{m}{n}})^k. \] (6)

The optimal number of the hashing functions \( k \) to reduce the false positives rate \( \psi \) is chosen according to the following equation Eq. (7):

\[ k = \ln(2) \cdot \frac{m}{n}. \] (7)

It is worth noting that increasing the number of hash functions \( k \) leads to higher computational overhead. Overall, for a Bloom Filter with \( m \) bits and \( k \) hashing functions, the time complexity is \( O(k) \) for the searching and inserting operations, while the space of the data structure is \( O(m) \).

In this paper, we selected to work with Compressed Bloom Filters, as proposed by in [35]. Compressed Bloom Filters are designed to minimize the false-positive rate \( \psi \), as a function of the Bloom Filter size \( m \), the number of elements \( n \), and the number of hashing functions \( k \), by leading to a reduction of the resulting bandwidth overhead. Let \( z \) the size of the compressed bloom filter, and \( E(p) = -p \log_2 p - (1-p) \log_2 (1-p) \) the binary entropy function, where \( p \) represents the probability that the value of each bit in the Bloom Filter is 0. The optimization consists in selecting the parameters \( m \) and \( k \) to minimize the false positive rate \( \psi \), according to Eq. (8)

\[ z \geq E(p) \cdot m. \] (8)

Assuming \( m = z \), and setting \( k = \ln(2) \cdot \frac{m}{n} \) as in Eq. (7), the relationship between the false-positives rate and the remaining parameters of the Bloom Filters shown in Eq. (6) rewrites as in Eq. (9).

\[ \psi = (0.6185)^\frac{m}{n}. \] (9)

According to the authors in [35], the above choice of \( k \) is the worst possible option, based on the dimension of the Bloom Filter \( m \). Despite further optimizations are possible, in this work, we adopted this worst configuration to provide an upper-bound on the resulting bandwidth overhead. At the deployment stage, it could be possible to reduce the bandwidth overhead further, if desired, without sacrificing security, at the expense of additional computational overhead (e.g., adding compression schemes, such as Huffman coding). This strategy could lead to further performance optimizations, but they are left as future work.

3 System and Security Requirements

The background study and system analysis carried out in the previous section allow us to identify a set of system and security requirements to be fulfilled by any security solution tailored to the AIS communication technology. A list of these and a brief discussion of their importance is provided below.

- **Authenticity of AIS Broadcast Messages.** The new security solution should be able to guarantee the authenticity of the received AIS messages. This means that it should be possible to uniquely establish a relationship between an AIS message and the entity that emitted this message.
- **Backward Compatibility.** Any security solution for the AIS technology should not require a mandatory migration to new hardware or software. This means that any legacy device that cannot be updated or does not require security features should continue to operate smoothly, without any degradation of the offered services. In other words, this requirement implies that any security update should work aside from the existing deployment, without modifying the current services.
- **Standard Compliance.** The AIS standard that is published today has been consolidated after years of modifications and amendments, and it would be very difficult (and lengthy) to modify it. Thus, the security solution to be developed must be compliant with the message format and the communication logic defined by the latest version of the AIS standard.
- **Openness.** The position and velocity data of vessels and ships worldwide are already used by many online websites and communities, offering advanced services over the gathered data. The openness of the above information allows also any receiver to monitor ships, and intervene immediately and efficiently in case of distress. Thus, to cause the minimum disruption, the new security solution should not compromise the openness of the AIS messages, still enabling interested parties to manage emergencies and create business opportunities over the data.
- **Robustness to Packet Loss Phenomena.** Especially in crowded scenarios such as ports and docking stations, multiple AIS transmission could collide, generating severe congestion phenomena. Thus, the security solution should guarantee that even under such a challenging network condition, the authenticity of each received packet could be established.
- **Minimal Bandwidth Overhead.** Due to the severe congestion that can take place in specific scenarios, the
security solution should require a limited amount of messages, not increasing the congestions of the communication link.

- **Flexibility.** Maritime scenarios can be characterized by very different operating conditions. For instance, when the vessel is offshore, only a few other vessels are in the neighborhood, and the RF communication link is not crowded. Conversely, when the vessel is in the port, the radio spectrum is congested, leaving only a few spaces for additional RF communications. At the same time, the AIS communication technology suggests the implementation of heterogeneous message rates, based on the operating speed of the vehicle. All these considerations should be carefully considered by any new security solution, in a way to adapt seamlessly to the specific scenario and operating conditions of the vessels.

The security solution we developed in this work considers together all the above-described requirements, and orchestrates a security framework specifically tailored to the requirements and operating conditions of maritime vehicles. Note that also other security requirements can be included, such as confidentiality, anonymity, and non-repudiation of AIS messages. On the one hand, we highlight that the fulfillment of such requirements is not in the scope of our paper. On the other hand, the proposed Auth-AIS scheme has been designed to be agnostic of the content and the type of the AIS messages, to be seamlessly integrable with any other standard-compliant security solution (such as the confidentiality scheme we proposed in [14]).

4 Scenario, Assumptions, and Adversarial Model

In this section, we describe the assumptions, the reference scenarios, and the adversarial model considered in this work. Section 4.1 introduces the reference scenarios as well as our assumptions, while the considered adversary model is detailed in Section 4.2.

4.1 Reference Scenarios

We assume a generic Class-B/SO vessel, equipped with an AIS transceiver integrated into the main bridge. Indeed, Class-A and Class-B/SO vessels are the only ones that can transmit AIS messages, and specifically, Class-B/SO transceivers are the ones that can use the channel for the least available time (3 slots). The vessel follows the recommendations of the AIS standard, accessing the wireless channel for RF transmission and reception operations according to its velocity and operational state. Being (at least) a Class-B/SO vessel, any transmission originated by the vessel cannot occupy the transmission medium for more than 3 consecutive AIS time-slots. We assume that the transmitting and the receiving vessels nearby are synchronized with an absolute reference clock, e.g., the one provided by the Global Positioning System (GPS). This is a reasonable and realistic assumption, since any AIS-equipped vessel also features a GPS receiving module, used to obtain precise location information. Moreover, we assume that the time is divided into time-slots of a fixed duration, in line with the AIS communication technology. Each time-slot lasts for 26.67 ms, and 2,250 time-slots form a slot-frame, that repeats over time. We also assume that the vessels are equipped with a GPS spoofing detection system, being able to detect (and possibly reject) GNSS spoofing attacks (e.g., IRIDIUM STL) [36].

Besides, it is worth noting that our work does not need any specific assumption on the specific AIS communication architecture. Indeed, being it either a direct RF communication link or a space-based AIS, the security services provided by our solution are not affected.

We also assume that each vessel is equipped with reasonable computational capabilities (such as the ones of a laptop). Given that we assume to work with (at least) Class B vessels, this assumption is in line with actual computational capabilities on board of commercial vessels. Furthermore, we assume that the vessel features a low-rate satellite Internet connection, such as the widespread IRIDIUM system, that can be optionally used to communicate with a Maritime Authority [37], [38]. This is also a realistic assumption, as nowadays many satellite-Internet providers such as IRIDIUM are certified GMDSS service providers, and thus they are frequently already integrated on-board. Satellite Communications (SATCOM) technologies such as IRIDIUM feature their own authentication and encryption schemes, used also for GPS spoofing evasion off-shore [39]. Therefore, we assume this communication channel as secure.

In this paper, we assume three reference scenarios regarding the position, velocity, and neighborhood of the transmitting vessel, as depicted in Fig. 2.

**Scenario 1: Offshore Vessel.** In this use-case, the transmitting vessel is located in the open sea, and it is traveling at the desired cruise speed. Thus, the vessel transmits broadcast reports at a high rate. In addition, being the scenario far from the mainland, there are few vessels in the neighborhood. Thus, the communication link is almost free, and many slots are available for the transmissions.

**Scenario 2: Vessel Moored in the Port.** In this use-case, the transmitting vessel is static and moored in the port. According to the latest AIS standard specification, it transmits position reports at a very low rate. At the same time, being in a crowded area such as a port, the communication link is crowded, and only a few time-slots per slot-frame are available for transmissions.

**Scenario 3: Vessel Moving in the Port.** In this use-case, the transmitting vessel is moving throughout the port. Thus, differently from the previous scenario, it transmits...
broadcast position reports at a high rate. However, being in a crowded area, the communication link is dense of concurrent transmissions, with only a few slots per slot-frame available for transmissions.

The qualitative description of the main system features of the above-described scenarios is summarized in the following Table 2. We notice that each scenario is characterized by a unique combination of system features, thus requiring a dedicated security solution.

Finally, we highlight that the definitions of the above scenarios allow defining operational requirements applicable only when the vessel is transmitting AIS packets. On the reception side, any vessel can correctly receive, decode, and authenticate packets emitted from any other vessel, independently from the specific scenario where the transmitting and the receiving vessels are operating. As will be discussed in Section 7.3, this inter-scenario interoperability can be achieved by inserting a simple security level indicator, specifying the security level currently in use by the transmitting vessel.

### Table 2: Qualitative System Features of the Scenarios Assumed in This Work

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Description</th>
<th>Reporting Congestion Rate</th>
<th>Congestion Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 1</td>
<td>Transmitting vessel offshore in the port</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Scenario 2</td>
<td>Transmitting vessel moored in the port</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Scenario 3</td>
<td>Transmitting vessel moving in the port</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

4.2 Adversary Model

The adversary assumed throughout this paper is characterized both by passive and active behavior. We assume that the adversary is equipped with an AIS transceiver module, that can be either a regular AIS handheld device or an SDR, operated through a laptop running an SDR-compatible software tool, such as GNURadio [40]. We assume that the adversary is a global eavesdropper, able to detect and decode any message exchanged on the two AIS communication channels. This feature allows the adversary to immediately obtain any information exchanged explicitly within the AIS messages, such as the MMSI of the vessels, their location information, and any other detail. Moreover, we assume that the adversary has also transmission capabilities. Thus, it can replay messages acquired at a given time, and it can inject new fake messages on the AIS communication channel. These messages report the MMSI of a legitimate vessel (acquired through eavesdropping on the channel), and they formally adhere to the AIS communication technology standard. Therefore, the forged messages are indistinguishable from legitimate AIS messages when processed by a regular AIS receiver. Furthermore, we also assume that the adversary can modify messages in transit on the communication channel, by performing selective jamming attacks. Thus, it can selectively cancel messages from the communication channel. In addition, the adversary can also change the message order, by first jamming and later replaying messages. This can be achieved since regular AIS broadcast messages do not include any sequencing mechanism, and they are ordered on the receiver only looking at the related time of arrival. We also assume that the adversary is mobile, i.e., it moves with the target vessels, being active for a significant time — the attack is not time-constrained.

The adversary’s goal is to have its forged (spoofed) messages to be accepted as legitimate, i.e., recognized by the receiver as if a legitimate AIS transceiver emitted them, and the importance of such attacks is emphasized by recent media reports such as [41]. We stress that the aim of the adversary considered in our work is not to carry out DoS attacks and not to eavesdrop on AIS messages or track AIS-equipped ships. Therefore, other traditional attacks connected to the usage of wireless communication channels, such as eavesdropping, tracking, and jamming, are not considered in our adversary model and are out of the scope of this paper.

At the same time, being Auth-AIS standard-compliant, any solution addressing such attacks in a standard-compliant way could be integrated seamlessly, without further adaptation. For instance, the protocol we proposed in [14], working on unicast binary messages and not on broadcast messages, can be integrated in parallel with the solution proposed here.

5 AIS Broadcast Authentication

In this section, we provide the details of Auth-AIS, the secure and flexible authentication protocol designed in this paper. Specifically, Section 5.1 provides the details of the system architecture and the protocol extension required by our proposal, while Section 5.2 provides the details of the operations and the message flow of Auth-AIS.

5.1 System Architecture and Protocol Extension

The system architecture assumed by Auth-AIS involves the following entities:

- **Maritime Authority.** It is a Trusted Third Party (TTP), whose role is to assign cryptography materials to the vessels. Besides, this authority is assumed to be always available online, ready to provide some public parameters of the vessels to the requesting entities. It is contacted at least once by each vessel during the setup phase (through standard Internet connection), and vessels could further contact it asynchronously during the online phase (via SATCOM links). In the real world, this role is already played by the IMO.

- **Transmitting Vessel.** It is a generic regular vessel, equipped with an AIS transceiver, and broadcasting position messages on the AIS communication channels. We also assume that it is equipped with our software solution, emitting dedicated messages to authenticate the broadcasted information.

- **Receiving Vessels.** It is a generic regular vessel, which we assume receives AIS messages from the transmitting vessel. Besides, we assume that it is interested in verifying autonomously the authenticity of the received broadcast messages.

We recall that Auth-AIS requires the transmission of dedicated messages from the vessel, including the cryptography material to be used to authenticate the regular AIS
messages. To this aim, Auth-AIS uses the Message Type 8, a broadcast message type that can be used by any application running on top of AIS to send dedicated broadcast messages, via a specific Function ID. Therefore, all the information used for the authentication of the broadcast are included as the payload of this message, with a Function ID set to the value 51 (left unused on purpose by the standard, to allow the crafting of special messages, like ours).

We also note that the messages emitted by the transmitting vessel do not contain any fragment ID. Indeed, according to the AIS standard specification, a transmitter keeps transmitting continuously for a maximum number of 3 consecutive AIS time-slots, without any interruption of the message flow. Thus, on the receiving side, the message appears to be a unique message flow, irrespective of the number of time-slots employed to be delivered. When more messages are required, they are ordered only based on their time of arrival at the receiver.

### 5.2 Auth-AIS Message Flow

The goal of Auth-AIS is to allow any AIS-compatible receiver to establish the authenticity of AIS broadcast packets emitted by a transmitting vessel. To this goal, the protocol is structured over two phases: the Setup Phase and the Online Phase, described in the following.

1. **Setup Phase.** This phase is executed by the vessel once per journey, at the boot-up of the system. Specifically, as depicted in Fig. 3, it consists of a one-shot interaction between the vessel and the maritime authority, aimed at initializing dedicated cryptography materials to the vessel.

   Specifically, on receiving a request by a generic vessel (identified through the MMSI ID), the maritime authority computes and provides the following values to the vessel:

   - **Master Key** $K_M$. It is a bit-string of size $|K_M| = n$, assigned uniquely to the requesting vessel for the whole duration of the journey. Note that a new master key is assigned to the vessel for each journey.
   - **$n$**. It indicates the size of the hash chain to be used for the TESLA protocol. Note that it represents also the (maximum) number of keys that will be generated locally by the vessel before asking for a new Master Key.
   - **$\phi$**. This is a global threshold value, to be used by the vessels to switch between the different security configurations.
   - **$S$**. Size of the Bloom Filter to be used for the probabilistic security configuration.
   - **$N_H$**. Number of hash functions to be used for the probabilistic security configuration.

   Upon receiving these values, the vessel first stores them for local use. Then, according to the TESLA protocol, the vessel can generate the local Root Key namely $K_0$, according to the following Eq. (10).

   \[
   K_0 = H(H(\cdots (H(K_M)) \cdots)) = H^n(K_M),
   \]

   where $H$ is a generic hash function, and the notation $H^n(K_M)$ refers to the application of the hashing operation $H$ over the key $K_M$ for $n$ times, recursively. Note that the generation of $K_0$ is done in parallel also by the Maritime Authority. Then, the Maritime Authority publishes the following parameters:

   - **MMSI ID**: it is the unique identifier of the vessel, in line with the AIS standard specification.
   - **$K_0$**: it is the root key of Auth-AIS for the specific vessel.
   - **$t_0$**: it is the starting time of Auth-AIS on the specific vessel.

   These values are assumed to be always available online, to be queried by any AIS transceiver interested to verify the authenticity of the broadcast messages emitted by the specific vessel.

2. **Online Phase.** This phase is executed at run-time, during the regular operations of any transmitting vessel. Overall, two main security configurations are provided by our protocol. The first one, namely Deterministic Security Configuration, provides a deterministic security level to the AIS messages, by integrating the TESLA protocol in the AIS communication technology. Despite being a deterministically secure solution, this configuration comes with a non-negligible bandwidth overhead, that could be overwhelming in congested scenarios. To cope with this limitation, our framework allows selecting a Probabilistic Security Configuration, where the deterministic security guarantees provided by TESLA are coupled with the probabilistic security features provided by the Bloom Filters.

In line with the latest AIS standard specification, the first step of the online phase consists of the passive listening of the AIS communication channels, for a fixed time. This phase is meant to estimate the number of vessels in the neighborhood of the transmitting vessel, as well as to have a view of the number of free time-slots available for transmission. Let us assume that $L$ are the total number of time-slots sensed by the vessel, and $R$ are the number of free time-slots, i.e., the time-slots where no transmissions are detected. For ease of discussion, we assume that the output of the channel sensing is a value $r = \frac{R}{L}$, representing the relative frequency of the free time-slots, as sensed by the transmitting vessel. Note that this operation is fully in line with the latest AIS specifications. Indeed, the AIS standard recommends a periodical sensing of the RF communication channel to obtain the information about the time-slots available for transmission.
Then, the pieces of information extracted from the channel sensing phase are combined with locally-available data, about the velocity and position of the vessel. The result is a specific security configuration of the protocol, selected in a way to achieve the best trade-off between security, bandwidth, and authentication delay. Recall that each vessel owns a threshold value $\phi \in [0, 1]$, as received by the Maritime Authority during the setup phase. This value represents the threshold to be used to decide which security configuration to adopt. Thus, each vessel compares the value of $r$ related to the real-time congestion degree of the channel with the threshold value of $\phi$. If $\phi < r$, the Auth-AIS protocol is set up with the Deterministic Security Configuration. Otherwise, if $\phi \geq r$, the Auth-AIS protocol is set up with the Probabilistic Security Configuration.

**Deterministic Security Configuration.** The Deterministic Security Configuration provided by Auth-AIS is based on the integration of the well-known TESLA protocol within the AIS communication technology, in a standard-compliant and backward-compatible fashion. The message flow of this configuration is depicted in the following Fig. 4.

The protocol works according to the following steps:

- Let us assume a generic time-slot $t_i$. At the beginning of the time-slot, the transmitting vessel $A$ computes the key for this time-slot, according to the TESLA protocol, as in the following Eq. (11):

  \[
  K_i = H^{n-i}(K_M).
  \]  

- Then, for each message $j$ transmitted in the time-slot $i$, namely $m_{i,j}$, the transmitting vessel $A$ pre-computes the Auth-AIS digest, namely $h_{i,j}$, according to the following Eq. (12):

  \[
  h_{i,j} = HMAC(m_{i,j}, K_i),
  \]

where the notation $HMAC(m, K_i)$ refers to a generic Hashed Message Authentication Code (HMAC) operation on the message $m$ using the key $K_i$. All these digests are locally stored.

- At the expiration of the time-slot $i$, i.e., at the beginning of the time-slot $i + 1$, the transmitting vessels $A$ broadcasts the Auth-AIS broadcast authentication message (message type 8), containing the key for the previous time-slot $i$, namely $K_{i-1}$, and all the digests of the messages previously computed, namely $h_{i,j}$.
- On the reception side, a generic receiving vessel $B$ first verifies that the received Auth-AIS digests $h_{i,j}$ have been created using the key $K_i$. To this aim, the verification of the following Eq. (13) is carried out:

  \[
  h'_{i,j} = HMAC(m_{i,j}, K_i) \Rightarrow h_{i,j}.
  \]  

If Eq. (13) is not verified, the message $m_{i,j}$ is immediately discarded as not authentic.

- Conversely, if there is at least one message such that Eq. (13) is verified, the Vessel $B$ can use the **Root Key** to authenticate the message. If the vessel $B$ does not possess the root key, it can contact the Maritime Authority, to receive the **Root Key** associated with the transmitting Vessel $A$. Note that this operation is only required once per transmitting vessel. It could be executed also before the departure of the vessel, and in case the vessel $B$ departed after the vessel $A$, the root key of the vessel $B$ is included in the list of root keys downloaded by the vessel $A$ from the Maritime Authority. Indeed, note that not all the keys for all the vessels should be downloaded, but only the ones of the vessels that are supposed to be on the same path of the vessel $B$. Moreover, as anticipated in Section 4.1, we assume the satellite communication link between the vessel and the IMO is secure, as it uses proprietary authentication and encryption protocols [39].

- On the reception of the root key from the maritime authority, the vessel $B$ can verify that the key $K_i$ used to authenticate the above messages has been computed starting from the root key stored on the maritime authority. To this aim, the following Eq. (14) is executed:

  \[
  K_0 = H^0(K_i).
  \]

If the above equation is verified, all the messages $m_{i,j}$ verifying also Eq. (13) are authenticated. Otherwise, all those messages are discarded as not authenticated.

We note that the above configuration of the Auth-AIS protocol is mostly in line with the system requirements defined in Section 3. First, it achieves a standard-compliant message authentication, without compromising the capability of legacy AIS transceivers to still receive clear-text (unauthenticated) AIS messages. At the same time, the scheme is also robust to packet loss phenomena. Indeed, a message loss affecting one packet or the digest of this packet does not affect the possibility to correctly authenticate other AIS messages.

Conversely, a weakness of this configuration is the required message overhead. Indeed, for each clear-text AIS
message $m_i$, a digest $h$ should be transmitted, as well. To overcome this limitation in severely congested scenarios, the proposed Auth-AIS scheme could be configured also with a Probabilistic Security Configuration, described below.

**Probabilistic Security Configuration.** The Probabilistic Security Configuration is adopted when a limited amount of free time-slots are available in the RF channel (typically, in a port). The message flow of this configuration is depicted in Fig. 5.

Specifically, the following steps are executed:

- Similarly to the previous description, let us assume a generic time-slot $t_i$. At the beginning of the time-slot, the transmitting vessel $A$ computes the key for this time-slot, according to the TESLA protocol, as in the Eq. (11) previously defined.

- For ease of discussion, we assume that starting from the time-slot $t_i$, the transmitting vessel $A$ deliver $N$ broadcast AIS messages, i.e., $\{m_{i,1}, m_{i,2}, \ldots, m_{i,N}\}$, where $N$ is a pre-defined number of messages. Note that, in line with the operation principles of the Bloom Filters, $N$ should be carefully set up to deal with the probability of false positives. However, we will provide in Section 7 a thorough study of the logic driving the choice of the value $N$. According to our protocol, each of these messages is temporarily stored in a local buffer.

- For each delivered message $m_{i,j}$, the transmitting vessel $A$ maps the message in the Bloom Filter $b_i$, according to the following Eq. (15).

\[ b_i = BF(m_{i,j}). \]  

(15)

Note that each single clear-text AIS message is mapped in the Bloom Filter as soon as it is delivered. This strategy, conversely to the one where all the elements are mapped together at the time when the message $m_{i,N}$ is delivered, allows avoiding possible delays due to multiple consecutive element mapping operations.

- Without loss of generality, let us assume that the message $m_{i,N}$ is delivered during the time-slot $i$. Then, at the beginning of the time-slot $i+1$, i.e., at the end of the time-slot $i$, where the message $m_{i,N}$ was transmitted, the transmitting vessel $A$ computes the Auth-AIS digest, considering as the input the Bloom Filter string $b_i$, as in Eq. (16)

\[ h_i = HMAC(b_i, K_i). \]  

(16)

- Then, the transmitting vessel $A$ delivers in broadcast the Bloom Filter $b_i$, the key $K_i$, and the digest $h_i$, through dedicated Auth-AIS broadcast authentication message (Message Type 8). Note that two options are possible: (i) delivering all the above materials together in a single AIS message, or (ii) delivering the Bloom Filter in one message and the key and the digest in another message. In Section 7 we will investigate the advantages of both the strategies.

- On the reception of the above-described values, the vessel $B$ first verifies that the received Auth-AIS digests $h_i$ have been created using the key $K_i$. To this aim, the verification check of the following Eq. (17) is executed

\[ h'_i = HMAC(h_i, K_i) = b_i. \]  

(17)

If Eq. (17) is not verified, the Bloom Filter $b_i$ and all the messages $m_{i,j}$ associated with this Bloom Filter are immediately discarded as not authentic.

Conversely, if Eq. (17) is verified, the Vessel $B$ can proceed to evaluate the authenticity of the cryptography materials. Similarly to the previous configuration, the **Root Key** of the vessel $A$ could be either asked via the Internet to the Maritime Authority, or be downloaded before the departure of the vessel (if the vessel $B$ departed after the vessel $A$), based on the intended path.

- On the reception of the **Root Key** from the maritime authority, the vessel $B$ can verify that the key $K_i$ used to authenticate the above messages have been computed starting from the **Root Key** stored on the maritime authority, applying the Eq. (14) previously defined. If the equation is verified, the Bloom Filter is considered as **authentic**, i.e., delivered by the vessel $A$.

- Then, for each message $m_{i,j}$, $j \in [1, N]$, the receiving vessel $B$ checks if the message is in the Bloom Filter, by applying the **membership query** operation defined in Section 2.4. Each message verifying the **membership query** check is considered as authentic, while any message not satisfying the above check is discarded as not authentic.

We highlight that the **probabilistic security configuration** allows reducing significantly the bandwidth overhead of Auth-AIS compared to the **deterministic security configuration**. Indeed, the digest of Auth-AIS is transmitted only once every $N$ messages, along with the Bloom Filter. However, this
improvement in the message overhead is obtained at the expense of a non-zero probability of false positive, i.e., a non-zero probability that a message could be considered as authentic, despite being forged. In Section 7 we will provide more details on the specific tuning of the parameters of the Bloom Filter, allowing to have a very limited false-positives rate. Besides, note that the message flow of the protocol does not change when the $N$ messages are delivered in more than a single time-slot. Indeed, in this case, the key used to generate the digest of the Bloom Filter is computed based on the time when the message $m_{i,j}$ is delivered, and all the previous $N−1$ messages are authenticated using this key, irrespective of the time-slot where they have been delivered.

We highlight that giving each vessel the right to select the threshold value $\phi$ autonomously could create potentially undesired situations, where selfish vessels could select the Deterministic Security Configuration despite the currently high congestion rate of the AIS communication channel, severely degrading the throughput of the channel. Therefore, the Maritime Authority could impose the value of $\phi$, i.e., it could indicate to the vessels in which conditions it is mandatory to use the Probabilistic Security Configuration. As will be clear later on, a vessel can fine-tune the specific overhead of Auth-AIS by selecting a suitable security level.

Finally, we remark that, as the reader can verify from the above description, the authentication features are added on top of the existing AIS standard specification, in a standard-compliant and backward-compatible fashion. Therefore, any AIS receiver that does not intend to verify the authenticity of the messages or does not feature the software update will continue to receive AIS messages. However, it will not authenticate them.

6 Security Considerations

We highlight that the security of Auth-AIS relies on the deterministic and probabilistic security features offered by well-known building blocks, such as TESLA and the Bloom Filters, whose security has been extensively proved in the past [32], [34], [42], [43].

Overall, the following security features are jointly offered by Auth-AIS.

Identification and Rejection of Forged AIS Information Messages. Auth-AIS can identify and reject forged AIS information messages efficiently, thanks to the consistency checks performed by any AIS receiver when the key and digest of the reference time-slots are disclosed. When Auth-AIS is configured with the Deterministic Security Configuration, assuming the adversary injects $J$ forged AIS information messages and $J$ forged digests, Auth-AIS enables a straightforward identification of the attack. Indeed, the adversary cannot know in advance the time-slot key $K_i$ used by the legitimate transmitter to authenticate all the messages delivered in the time-slot $t_i$. Therefore, the adversary will use a key $K'_j$ that is different from the legitimate one, $K_i$. As soon as the legitimate key is revealed, in the time-slot $t_{i+1}$, any receiver could be able to verify Eq. (14), and authenticate all the legitimate messages, while discarding all the others.

Similarly, when Auth-AIS is configured with the Probabilistic Security Configuration, the legitimate time-slot key $K_i$ will enable the verification of Eq. (17), and thus of the legitimate Bloom Filter. Therefore, only the messages satisfying the membership check over the legitimate Bloom Filter $b_i$ will be authenticated, while the others will be discarded.

Identification and Rejection of AIS Replay Attacks. Auth-AIS can identify and reject replayed AIS information messages, provided that they are replayed in a different time-slot than the one where the legitimate messages were delivered. Let us assume that Auth-AIS is configured with the Deterministic Security Configuration and that the adversary in the time-slot $t_i$ replays messages that were sent out by a legitimate vessel in a time-slot $t_j$, with $j < i$. Then, any AIS receiver will be able to detect the replayed messages when the key used to authenticate the messages in the time-slot $t_i$ is disclosed, i.e., in the time-slot $t_{i+1}$. Indeed, by applying Eq. (13) using the key $K_j$ on the replayed message, the equation will not be satisfied, leading to the rejection of the message. Similarly, using the Probabilistic Security Configuration, there will be only one Bloom Filter satisfying Eq. (17) using the correct key $K_i$. By applying the membership check on this Bloom Filter string using the forged message, the message will be discarded with high probability (we refer the readers to Section 7 for the details of the false-positive rate achieved by Auth-AIS).

Identification and Rejection of Key, Digest, and Bloom Filter Replays and Messages Injection. Auth-AIS can reject also forged key, digest, and Bloom Filter messages. Indeed, independently from the selected security configuration, the security of Auth-AIS is based on the impossibility for the adversary to obtain the key of a slot $t_i$ from the key of a slot $t_j$, where $i < j$. To have success, the adversary should be able to reverse the cryptographic hash function in Eq. (14), which is computationally impossible by construction for a generic hash function [44]. Therefore, assuming the master key $K_{\text{master}}$ is kept secret by the vessel and the maritime authority, the legitimate vessel will be the only one able to obtain the time-slot key $K_i$ before its public disclosure (in the time-slot $t_{i+1}$), and an adversary will not be able to compute it in advance.

Robustness to Partial Key and Digest Message Loss. Auth-AIS can also overcome efficiently partial key and digests message losses. Let us assume that, in a reference time-slot $t_i$, either the key or the digest of Auth-AIS are not received by a generic AIS receiver. This event prevents the instantaneous verification of all the messages sent in that time-slot. However, this temporary out-of-service of Auth-AIS could be immediately recovered in the following time-slot. Indeed, in the new time-slot $t_{i+1}$, the related time-slot digest and key (and, eventually, the Bloom Filter) will be delivered again in dedicated messages, enabling the verification of the messages transmitted in this time-slot. We also highlight that, in case only the key of the time-slot $t_i$ is lost, but the digest $h_i$ is received, the messages sent in the time-slot $t_i$ can be authenticated as soon as the key $K_i$ of one later time-slot is received and verified. Indeed, assuming $j = i+1$, any AIS receiver could easily verify that $K_i = H(K_j)$, obtaining the lost key and verifying, with a small delay, all the messages transmitted in time-slot $t_i$.

Protection against Digest and Key Forgery. In Section 7.1 we will evaluate the bandwidth overhead of Auth-AIS in the different security configurations, while varying the size of the cryptographic digest and the Bloom Filter. These variables are directly connected with the security level of Auth-AIS.
AIS, as well as its protection against digest and key forgery. Overall, reducing the size of the digest and key reduces, in turn, the bandwidth overhead of the security solution. However, this strategy reduces the security level of the protocol as well, as it increases the probability that the adversary could find a digest colliding with the legitimate one [45]. Therefore, the longer the digest and the longer the key, the higher the security level of the protocol and its protection against digest and key forgery. Besides, note that any key used in the protocol has no validity for the adversary if found after the expiration of the related time-slot (a time-slot key $K_i$ is used to authenticate only messages sent in that time-slot). Therefore, the task of the adversary is further complicated by the limited time available for the attack, usually bound by the duration of the travel. The interested readers can refer to [42] and [43] for the formal evaluation of the security properties offered by TESLA, while [34] provides details on the security usage of Bloom Filters.

The performance evaluation reported in the following section will provide evidence that Auth-AIS can be configured always with the minimum digest size of 160 bits and a false-positives rate less than $2^{-40}$, while requiring only a limited bandwidth overhead.

7 PERFORMANCE ASSESSMENT

In this section, we provide a thorough performance assessment of Auth-AIS. Specifically, Section 7.1 provides the details of the tuning of the security configurations of the protocol, while Section 7.2 includes a comparison between Auth-AIS and a competing state-of-the-art solution. Finally, the implementation of Auth-AIS in the GNURadio ecosystem has been described in Section 7.3.

7.1 Tuning of the Security Configurations

In this section, we investigate the optimal tuning of the parameters of Auth-AIS. Specifically, our investigation aims at identifying the values to be assigned to the different parameters of Auth-AIS, to best trade off between the security level, the bandwidth overhead, and the authentication delay of the protocol.

In the following, we considered the requirements imposed by the latest AIS standard specification, regarding the duration of the time-slots and the access to the wireless channel by a generic AIS transceiver. Specifically, given that our solution is meant to be operated (at least) by any Class-B/ SO AIS transceiver, in line with the standard, we imposed that any broadcast authentication message (message type 8) does not last for more than 3 consecutive AIS time-slots. In turn, this requirement leads to a maximum payload size for any broadcast authentication message of 66 bytes. Considering to dedicate 1 byte to the protocol metadata, 65 bytes are available for security-related values. Moreover, we assume that each clear-text AIS message lasts for 1 time-slot, in line with the standard broadcast position-report AIS message (message type 1).

As an evaluation metric, we consider the bandwidth overhead, defined as the ratio between the number of AIS time-slots to send the broadcast authentication message(s) and the total number of authenticated AIS time-slots (evaluated in percentage). For instance, if 3 clear-text AIS time-slots are authenticated through a message lasting 2 time-slots, the bandwidth overhead is $\text{overhead} = \frac{2}{3} \times 100$.

Based on the above considerations, in the following, we evaluate the tight relationship existing between the bandwidth overhead, the security level, the authentication delay, and the false-positives rate (if involved) of all the configurations of Auth-AIS, to identify the configurations that best fit all the requirements and scenarios previously defined (see Sections 3 and 4.1).

We first focus on the Deterministic Security Configuration. We fixed the size of the time-slot key to 16 bytes, and we evaluated the bandwidth overhead of this configuration of the protocol by varying the size of the digest. The results are reported in Fig. 6. We notice that the higher security level can be achieved using a digest of 49 bytes, requiring 3 additional time-slots for each AIS message and an overhead of 75 percent (3 over 4 AIS time-slots are dedicated to authentication values). This overhead is acceptable in the case of the Scenario 1 described in Section 4, where many time-slots are free to use. However, when the channel is congested (typical situation of a port), this configuration becomes highly inefficient, as it further congests the communication link. This is the reason driving the introduction of the Probabilistic Security Configuration.

For the Probabilistic Security Configuration, we evaluate the offered security guarantees by considering jointly the size of the cryptographic digest and the false-positives rate achieved through the Bloom Filter. As for the Bloom Filter, we define as acceptable only false-positives rates less than $2^{-40}$, i.e., characterized by (approximately) 1 false-positive event over $2^{40} \approx 10^{12}$ messages.

We notice that, considering the constraints previously introduced, the Probabilistic Security Configuration can be set up according to two different options. The first one, referred to as Option 1 in the following, consists of transmitting the Bloom Filter, the key, and the digest of Auth-AIS together in the same AIS message. The second one, referred to as Option 2 in the following, consists of transmitting the Bloom Filter in a single AIS message, while the key and the digest are transmitted in another AIS message. Note that each message takes a maximum number of 3 consecutive AIS time-slots.

Assuming the Option 1, we first investigated the security performance that can be achieved using this protocol configuration. Specifically, we investigated the relationship between the false-positives rate and the bandwidth overhead, by considering different sizes of the cryptographic digest. The
results of this study are reported in Fig. 7. We stress that the longest possible size of the digest we can include in this option is 384 bits, as at least 1 byte should be dedicated to the Bloom Filter, 16 bytes are allocated for the key and 1 byte is for protocol metadata (summing up to 66 bytes).

We notice that, under the constraints previously introduced, there are a few protocol configurations that can reduce the bandwidth overhead, while being still able to guarantee a false-positives rate below 2\textsuperscript{-40}. Considering the digest size of 256 bits, the configuration deploying the Bloom Filter allows having longer digests, at the expense of a reduced increase in the bandwidth overhead. In turn, a longer digest results in an increased security level. At the same time, more messages are needed to carry out the authenticity check. Indeed, to verify one of the 9 plain-text messages, the receiver has to wait for the transmission of a maximum number of 8 additional messages. These two configurations of the protocol are the most suitable ones in this scenario, assuming two reference sizes of the cryptographic digest (N = 2 elements in the Bloom Filter) with a single broadcast authentication message requires a message overhead of the 40 percent. At the same time, selecting a digest size of 160 bits and N = 4 elements in the Bloom Filter, Auth-AIS can achieve a false-positives rate slightly below 2\textsuperscript{-40}, with a bandwidth overhead of 42.86 percent. All the other configurations are either too bandwidth-greedy or they cannot guarantee the desired minimum false-positive rate. We notice that, under the requirements previously-defined, it is not possible to use securely the longest digest size (384 bits).

This configuration of Auth-AIS is particularly suitable to be used in the Scenario 2 defined in Section 4. We recall that, when the vessel is moored in the port, the reporting rate is low and, conversely, the congestion rate is high. Therefore, the suitable security solution should be able to provide messages authenticity both with a limited bandwidth overhead—due to the congestion—and with a low authentication delay. For instance, the configuration using a digest size of 256 bits computed over 2 AIS messages results in an overhead of 40 percent and a delay of only one message. Similarly, the configuration using a digest size of 160 bits over 4 clear-text AIS messages results in an overhead of 42.86 percent, slightly increasing the delay to up to 3 messages.

To find a suitable security configuration for the Scenario 3, we focus now on the Option 2 of the probabilistic security configuration. We recall that in this configuration the Bloom Filter is transmitted in a separate AIS message than the one used for delivering the key and the digest.

Fig. 8 reports the bandwidth overhead of this configuration against the false-positive rate that can be achieved in this scenario, assuming two reference sizes of the cryptographic digest, i.e., the minimum of 160 bits (minimum acceptable digest size, as specified in [45]) and the maximum of 392 bits (49 bytes, as previously motivated). We note that considering a digest size of 256 bits leads to the same results obtained for 392 bits.

Considering the digest size of 160 bits, we notice that the configuration closest to the threshold of 2\textsuperscript{-40} is the one where 9 elements (i.e., 9 clear-text AIS messages) are authenticated together. In this case, the bandwidth overhead amounts to 35.71 percent. Instead, considering a digest size of 320 bits and a number of elements equal to 9 in the Bloom Filter, the configuration closest to the false-positive rate threshold achieves a bandwidth overhead of 40 percent. Therefore, using separate messages for the digest and the Bloom Filter allows having longer digests, at the expense of a reduced increase in the bandwidth overhead. In turn, a longer digest results in an increased security level. At the same time, more messages are needed to carry out the authenticity check. Indeed, to verify one of the 9 plain-text AIS messages, the receiver has to wait for the transmission of a maximum number of 8 additional messages. These two configurations of the protocol are the most suitable ones in the Scenario 3. Indeed, when applied in this scenario, the last two configurations achieve little bandwidth overhead and reduced authentication delay (the reporting rate is high), meeting all the desired requirements. Additional message overhead reduction is possible by increasing the authentication delay. The following Table 3 summarizes the investigation conducted in this section, reporting the optimal security configuration to be used for each scenario.

### 7.2 Comparisons With Existing Approaches

This section compares the performance of Auth-AIS against other solutions available in the literature, sharing the same system and security objectives discussed in Section 3.

First, we compare Auth-AIS and the existing authentication solutions, discussed in Section 2.2 of the manuscript. The comparison is summarized in Table 4. We notice that the proposals in [15], [16], [17], and [18] are neither standard-compliant nor backward-compatible. Therefore, even if their message overhead is 0, their integration into existing deployments would require extensive hardware modification, incurring delays and high costs. To show that standard compliance is not the only advantage of our solution, we take a step further, evaluating the overhead of customized
standard-compliant versions of the solutions in [15], [16], [17], and [18]. For instance, for the proposal by the authors in [15], instead of embedding the public key signature into the same AIS packet (not standard compliant), we evaluated the overhead generated by the delivery of the signature into a dedicated AIS packet (standard compliant).

We highlight that, when integrated into AIS in a standard-compliant and backward-compatible fashion, the above-discussed proposals would require a single dedicated AIS packet for each message, with a 50 percent resulting security overhead. Similar considerations also apply for the proposal by the authors in [19]. Although the proposal is inherently standard-compliant, it requires at least a new message for each AIS broadcast packets, which introduces overhead, being hard to combine with Auth-AIS. We also remark that none of the previous solutions allowed for the switch of the configuration needed to adapt to commercial vessels’ heterogeneous operating conditions.

Conversely, Auth-AIS is standard-compliant and backward-compatible, and can adapt its overhead based on the vessels’ operating conditions. Moreover, its worst-case overhead varies in the range [35 – 75] % and can be tuned by the vessels autonomously. Therefore, even considering standard-compliant versions of the competing solutions, Auth-AIS is still the most efficient. Note that further bandwidth reduction is possible, e.g., by accepting more false positives. For instance, considering, on average, one false positive every 128 years of continuous AIS transmissions, Auth-AIS would introduce only 30 percent of overhead.

As summarized in the previous discussion, to the best of our knowledge, the current literature is lacking peer-reviewed published security solutions specifically conceived for the AIS communication technology and satisfying, at the same time, all the security requirements previously described. Therefore, to provide further insights on the performance of Auth-AIS, we selected a current security protocol that, despite fulfilling all the system and security requirements, has not been proposed in the context of AIS. Specifically, we selected the SOS protocol proposed recently in [26] in the context of ADS-B avionic communications.

We highlight that ADS-B and AIS are very similar communication technologies. Indeed, similarly to AIS for vessels, ADS-B allows aircraft to broadcast identification, position, and velocity data, with the aim of information management and self-separation.

Similarly to Auth-AIS, the SOS protocol proposed in [26] leverages the TESLA cryptographic building block, but it extends it further, by computing the digest over the concatenation of $N$ consecutive packets. Using such a strategy, the bandwidth overhead is reduced consistently. However, despite the advantage in terms of bandwidth overhead, this strategy is highly vulnerable to packet loss phenomena. Indeed, if only a single packet out of $N$ is not received, all of them cannot be authenticated. To evaluate the advantage of Auth-AIS against this reference solution, we evaluated the impact of the packet loss on both the strategies. Specifically, we assumed packet losses are distributed uniformly at random on the wireless communication channel, and we evaluate the authentication rate, defined as the ratio between the number of authenticated AIS packets and the overall number of clear-text AIS packets to be authenticated. We assumed different packet loss percentages affecting the communication channel and a different number of collaborating receivers. The results reported in Fig. 9 are obtained each over 1,000 realizations of the communication channel. Note that, for the Auth-AIS protocol, we assume the configuration characterized by the least bandwidth overhead, i.e., the one where 9 AIS plain-text messages are authenticated using a digest of 160 bits and a Bloom Filter of 392 bits.

We first focus on the scenario with a single receiver, i.e., the case of a standalone receiving vessel equipped with only a single antenna. We notice that Auth-AIS always outperforms SOS, independently from the specific percentage of the packet loss affecting the communication channel. For instance, assuming that 20 percent of the packets are lost uniformly at random, Auth-AIS achieves an authentication rate of 0.385, while SOS achieves only 0.101. This result can be explained considering that in SOS it is enough to lose one message to have that all the messages in the same time-slot cannot be authenticated. In the same conditions, instead, Auth-AIS can still verify all the other messages. The advantages of Auth-AIS in terms of authentication rate are even more noticeable when more receivers are collaborating. This is the case, for instance, of a port. In this scenario, multiple AIS receivers can be spread in different locations; then, the received packets are forwarded to a central processing unit, where they can be analyzed together. When multiple receivers are involved, Auth-AIS still outperforms SOS. For instance, considering 4 collaborating receivers and 50 percent of mean packet loss, Auth-AIS achieves an authentication rate of 0.678, while SOS only the 0.391.

Although a few DoS attacks could be enabled by the deployment of Auth-AIS, such as high-rate messages injection or selective jamming, they would not stop the system but only cause a roll-back to the current state. Moreover, we remark that DoS attacks are not in the scope of this paper.

Overall, these results confirm the enhanced suitability of Auth-AIS for AIS, even in lossy scenarios, as it allows to
authenticate a higher number of messages compared to competing solutions.

7.3 Implementation Details

To demonstrate the feasibility and applicability of Auth-AIS for AIS communications, we set up a real proof-of-concept of the protocol, using compatible AIS transceivers. As per the hardware platform, we used the Ettus Research X310 SDR, featuring a UBX160 daughterboard [46]. This setup is a common and reasonable choice when implementing and testing protocols for communication technologies such as AIS, requiring hardware working on dedicated non-ISM frequencies [26]. To interface with the SDR, we use a Laptop Dell Inspiron 7577, equipped with 4 intel core i7-7700 HQ CPU working at 2.5 GHz, 16 GB of RAM, and 1 TB of SSD.

As for the software, we implemented Auth-AIS in the GNU Radio ecosystem [47], extending the AIS transmission module originally provided by the authors in [5]. Specifically, we included an application that dynamically executes the security operations required by Auth-AIS based on the reporting time and the congestion of the communication link, encapsulates the information in standard-compliant AIS messages, generates the bit-sequence, and instructs the radio to transmit the messages according to the AIS standard specifications. As for the hashing and HMAC operations, we used the SHA512 hashing algorithm, truncating the size of the output to the desired size. It is also worth noting that our implementation assumes that the duration of the TESLA time-slot is coincidental with the duration of the standard AIS time-slot, i.e., 26.67 ms. On the reception side, we used the gr-ais module available at [48], modified in a way to synchronize with the transmitting module and achieve 100 percent reliability in the radio operations. Fig. 10 shows our experimental testbed. This is an implementation of the AIS communication technology that is regularly used by hobbyists to receive AIS messages [48], that further contributes to demonstrating the seamless feature and ease of integration of our solution.

Based on the analysis presented in Section 7.1, the implementation of Auth-AIS includes 7 different security levels, that can be automatically selected based on the reporting rate and the velocity of the vessel. They are:

0) **No Security.** AIS communications are still performed in clear-text, without any authentication service;

1) **Deterministic Security Configuration,** Digest Size of 49 bytes, key size of 16 bytes, sent out for every AIS message (overhead = 75 percent);

2) **Deterministic Security Configuration,** Digest Size of 21 bytes, key size of 16 bytes, sent out for every AIS message (overhead = 66.67 percent);

3) **Probabilistic Security Configuration, Option 1,** Bloom Filter size of 17 bytes, digest Size of 32 bytes, key size of 16 bytes, sent out for every $N = 2$ AIS messages (overhead = 60 percent);

4) **Probabilistic Security Configuration, Option 1,** Bloom Filter size of 29 bytes, digest Size of 20 bytes, key size of 16 bytes, sent out for every $N = 4$ AIS messages (overhead = 42.86 percent);

5) **Probabilistic Security Configuration, Option 2,** Bloom Filter size of 65 bytes, Digest Size of 20 bytes, and key size of 16 bytes, sent out every $N = 9$ AIS messages (overhead = 35.71 percent);

6) **Probabilistic Security Configuration, Option 2,** Bloom Filter size of 65 bytes, Digest Size of 49 bytes, and key size of 16 bytes, sent out every $N = 9$ AIS messages (overhead = 40 percent);

Based on the number of free time-slots available for transmission, without modifying the configuration of Auth-AIS, a vessel can select the security level that best suits the actual operating conditions.

Overall, a possible selection strategy for the threshold value $\phi$, the security levels, and the detected congestion levels can be the one proposed in Table 5.

Note that the logic used for the adoption of a particular security level is such that, in the worst-case where all the vessels are using a single slot for AIS transmission broadcast, the maximum resulting congestion rate is approximately 0.9. In this way, new vessels entering an area have a notable chance to find a free buffer for their AIS transmissions.

For instance, when a relevant congestion rate on the AIS communication link is detected (such as in a crowded port), the vessel can opt for the Security Level # 5, using a Bloom Filter size of 65 bytes, Digest Size of 20 bytes, and key size of 16 bytes, sent out every $N = 9$ AIS messages, with a resulting overhead of 35.71 percent. Conversely, when off-shore, likely experiencing a low level of congestion, the vessel could set the Security Level # 1, characterized by a digest Size of 49 bytes, a key size of 16 bytes, sent out for every AIS message, with an overhead of 75 percent. This selection can be easily automatized and integrated within the AIS transceiver, based on the number of available free time-slots.

We remark that the logic of $\phi$ is to discriminate between the adoption of the deterministic and the probabilistic
security configuration. Therefore, adopting the strategy reported in Table 5, the choice is $\phi = 0.3$. However, each vessel could fine-tune the security levels and the intervals in Table 5.

On the receiver side, the receiving AIS vessel can easily realize the specific security level used by the transmitter, by decoding the corresponding Security Level field of the AIS packet with message type 8, as defined in Section 5.1. We also notice that the specific security level used by the transmitter does not affect the security level selected by the receivers. Indeed, any vessel can use the desired security level, based on the local vision of the neighborhood.

Overall, our implementation requires 80 KB of physical RAM and 2,132 KB of virtual RAM for the transmitter, while for the receiver it requires 148 KB of physical RAM and 0 bytes of virtual RAM. Fig. 11 provides an overview of the duration of the main operations of Auth-AIS on the host laptop. Note that only the security levels in the probabilistic security configuration have been considered, as they include all the hashing, element mapping, and membership check operations. We notice that each single hashing operation required by Auth-AIS requires 52.22 ns (with a 95 percent confidence interval range of $[\pm 0.7006\text{ ns}]$). Considering the Element Mapping and Membership Check operations we notice that they take different times, increasing with the security level. For instance, we notice that the Element Mapping operation requires 132.8 ns when Auth-AIS works according to the security level 3, while it increases to 401.6 ns when Auth-AIS is configured with the security level 6. The increase in the time required to complete these operations is due to the size of the bloom filter and the number of hashing operations required by the bloom filter (recall Eq. (7)). Indeed, while the bloom filter has a size of 17 bytes in the Security Level 3, it increases to 29 bytes in the Security Level 4, and to 65 bytes in the Security Levels 5 and 6, and the number of hashing functions vary accordingly, as described in Eq. (7). Overall, the maximum time required to perform these operations is always less than 500 ns, being many orders of magnitude smaller than the duration of an AIS time-slot. Note that an AIS transceiver not equipped with Auth-AIS would continue to operate regularly.

Finally, we remark that we released the source code of Auth-AIS in the GNURadio development toolkit as open-source, together with the clip of a demo showing the effectiveness of our solution [7]. The code can boost research on AIS communication technology, offering a ready-to-use basis for comparisons and further software development.

---

### TABLE 5

<table>
<thead>
<tr>
<th>Security Level</th>
<th>Security Overhead [%]</th>
<th>Detected Cong. Rate</th>
<th>Maximum Cong. Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>75</td>
<td>$&lt;0.15$</td>
<td>$\approx0.9$</td>
</tr>
<tr>
<td>2</td>
<td>66.67</td>
<td>$[0.15 - 0.23]$</td>
<td>$\approx0.9$</td>
</tr>
<tr>
<td>3</td>
<td>60</td>
<td>$[0.23 - 0.3]$</td>
<td>$\approx0.9$</td>
</tr>
<tr>
<td>4</td>
<td>42.86</td>
<td>$[0.3 - 0.48]$</td>
<td>$\approx0.9$</td>
</tr>
<tr>
<td>5</td>
<td>35.71</td>
<td>$&gt;0.55$</td>
<td>$\approx0.9$</td>
</tr>
<tr>
<td>6</td>
<td>40</td>
<td>$[0.48 - 0.55]$</td>
<td>$\approx0.9$</td>
</tr>
</tbody>
</table>

---

Fig. 7. Performance evaluation in real conditions

We deployed an AIS sensor in Doha, Qatar. The AIS sensor consists of an RTL-SDRv2 Software Defined Radio (SDR), connected to a Raspberry PI system. The RTL-SDR features a multi-purpose dipole antenna, that can efficiently receive signals in the bandwidth $[50 - 1.000]\text{ MHz}$ (thus being able to detect AIS wireless messages). On the Raspberry-PI, we installed the gr-ais module, that post-processes the AIS messages and decode them. Finally, the decoded data are uploaded to AISHub through a wired Internet connection. A screenshot of the typical coverage range of the sensor is shown in Fig. 12. Note that the location of the receiver is close to a touristic port.

We acquired data through the previously-described AIS receiver for a time window of two (2) weeks, and we stored all the data on a local database. Then, based on the reception timestamp of the AIS messages, we inferred the congestion level of the AIS communication link in multiple overlapping time windows of one minute, as described by the AIS standard. For our analysis, we considered the minimum, maximum, and median number of busy slots found in the reference observation time, and we evaluated the impact of each of the six security configurations of Auth-AIS in these reference scenarios. The results of our investigation are summarized in Table 6. Note that the minimum, median, and maximum number of busy slots found in our data are 1, 28, and 285, respectively. Moreover, note that for the evaluation of the congestion level after the inclusion of Auth-AIS, to enable a fair comparison among the different security levels, we considered different consecutive time-windows, where the specific congestion rates occur consecutively. For instance, for the Security Level # 3, we
Table 6

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.04</td>
<td>0.17</td>
<td>0.13</td>
<td>0.11</td>
<td>0.07</td>
<td>0.05</td>
<td>0.06</td>
</tr>
<tr>
<td>28</td>
<td>1.24</td>
<td>3.73</td>
<td>2.49</td>
<td>2.48</td>
<td>1.86</td>
<td>1.52</td>
<td>1.66</td>
</tr>
<tr>
<td>285</td>
<td>12.67</td>
<td>38.0</td>
<td>25.33</td>
<td>25.33</td>
<td>19.0</td>
<td>15.48</td>
<td>16.89</td>
</tr>
</tbody>
</table>

considered two time-windows of 1 minute, where each of the time-windows is characterized by a congestion level of 1, 28, and 285 busy slots over the total number of 2,250 available time-slots.

The results reported in Table 6 highlight that each security level of Auth-AIS is characterized by a different overhead. While the Security Level #1 is the most bandwidth-hungry configuration, the Security Level #5 allows reducing the overhead generated by security messages, at the expense of a slight increase in the authentication delay.

Overall, in combination with the Maritime Authority, a vessel could select the Security Level that best trades off authentication delay and congestion rate.

8 Conclusion

In this paper, we presented Auth-AIS, a secure, flexible, and backward-compatible broadcast authentication protocol for vessels AIS communications. Auth-AIS can be integrated into existing AIS deployments as a simple software update, not requiring any hardware modification. Integrating well-known building blocks, such as TESLA and Bloom Filters, Auth-AIS can provide the authentication of AIS broadcast trading off time-delay with bandwidth overhead, based on the requirements of the scenario. When configured with the Deterministic Security Configuration, Auth-AIS provides immediate authentication of AIS broadcasts using digests up to 49 bytes long, with a bandwidth overhead of 75 percent. When the communication channel is busy, Auth-AIS can be configured with the Probabilistic Security Configuration, reducing the bandwidth overhead of authentication processes down to 35.71 percent, guaranteeing a minimum acceptable security level equivalent to 80 bits, and a practically negligible false-positive rate ($2^{-40}$). Intermediate configurations of the protocol are also applicable, offering trade-offs between the security level, the message overhead, and the authentication delay. We also implemented a proof-of-concept of Auth-AIS, using the GNURadio ecosystem and the Ettus Research X310 hardware. The source code has been released as open-source, to allow Industry and Academia to further innovate in the field of secure maritime communications.

Through our work and the publicly-released open-source code of the proof of concept, we aim to communicate to the scientific community, the companies, and the industrial committee actively working on the development and standardization of the AIS technology, that is possible to provide a standard-compliant, backward-compatible, flexible, and secure solution for the AIS communication technology. Therefore, the integration of Auth-AIS into real AIS-equipped vessels is the final objective of our work, and it is a long-medium-term task left for future work.

Acknowledgments

The authors would like to thank the anonymous reviewers for their comments and insights, that helped improving the quality of the paper. This work was supported by the Qatar National Research Fund (a member of Qatar Foundation) under Grant NPRP-S-11-0109-180242. The information and views set out in this publication are those of the authors and do not necessarily reflect the official opinion of the QNRF.

References


Savio Sciancalepore received the bachelor’s degree, master’s degree, and PhD degree from Politecnico di Bari, Italy, in 2011, 2013, and 2017, respectively. He is currently an assistant professor with the Eindhoven University of Technology. From 2017 to 2020, he was a postdoc with HBKU-CSE-ICT. His research interests include security issues in Internet of Things and cyber-physical systems.

Pietro Tedeschi received the master’s degree (Hons.) in computer engineering from Politecnico di Bari, Italy. He is currently working toward the PhD degree with HBKU-CSE-ICT. He was a security researcher with CNIT, Italy, for the EU H2020 SymboleTo. His security research interests include UAVs, wireless, IoT, and applied cryptography.

Ahmed Aziz received the bachelor’s degree in electrical (telecommunication) engineering from MCS, NUST, Pakistan and the master’s degree in cybersecurity from HBKU, Qatar. His security research interests include wireless, IoT, network, and computer security.

Roberto Di Pietro is currently a full professor of cybersecurity with HBKU-CSE and a distinguished scientist with ACM. His research interests include distributed systems security, wireless security, and OSN security. In 2011–2012, he was the recipient of the Chair of Excellence from University Carlos III, Madrid. In 2020, he was the recipient of the Jean-Claude Laprie Award for having significantly influenced the theory and practice of Dependable Computing.