

Bugten, Brage Westby
Løkken, Christopher Collin

Security challenges of VDES and future VDES-based services

Master's thesis in Communication Technology and Digital Security

Supervisor: Bernsmed, Karin

Co-supervisor: Bour, Guillaume

June 2023

Bugten, Brage Westby
Løkken, Christopher Collin

Security challenges of VDES and future VDES-based services

Master's thesis in Communication Technology and Digital Security
Supervisor: Bernsmed, Karin
Co-supervisor: Bour, Guillaume
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology



Title: Security challenges of VDES and future VDES-based services

Students: Bugten, Brage Westby and Løkken, Christopher Collin

Problem description:

VHF Data Exchange System (VDES) is an emerging communication system to be deployed to ships in the coming years. Compared to its predecessor, Automatic Identification System (AIS), VDES adds more communication channels, increased bandwidth, two-way communication, and improved security, paving the way for several future digital services. Since VDES is intended as an extension of AIS, VDES needs to be compatible with older versions of AIS. It is well known that AIS is vulnerable to various attacks. VDES will empower Vessel Traffic Services (VTSs), improve safety onboard ships, and aid Search and Rescue (SAR) operations with increasing satellite coverage, crucially in polar areas. In this ecosystem, VDES is a key enabler according to the International Maritime Organization (IMO)'s E-navigation Strategy Implementation Plan (SIP), which aims at modernizing the flow of data and information between sea- and land-based actors in the maritime industry. In this recommendation, IMO have identified 16 maritime services and areas of improvement which will automate and optimize many of the manual processes which, in many instances, today, are handled using Very High Frequency (VHF) voice communication. Providing these services over VDES is a big shift and will require international cooperation between stakeholders and standardizing bodies. At the same time, this transition will potentially expose end-users and service providers to new cyber risks that previously were not feasible due to the underlying technologies that have been out of reach for attackers. By conducting an in-depth look at VDES, we also want to assess the possibility and difficulty of conducting already existing attacks on AIS, in VDES. Using a combination of the Cyber Kill Chain (CKC) and a Resource Cost Model (RCM), our thesis will attempt to model various attacks towards these services and their associated difficulty at different attack stages. These results will create a high-level risk assessment of future maritime services.

Approved on: March 23rd 2023

Main supervisor: Dr. Bernsmed, Karin, NTNU

Co-supervisor: Bour, Guillaume, SINTEF Digital

Abstract

A strong increase in vessels in the world's oceans and inland waterways has slowly become more demanding for maritime Information and Communication Technology (ICT) systems. VTSs, maritime service providers and international authorities are struggling to keep up with the growth, as systems such as AIS are reaching their capacity limits, challenging their ability to monitor and manage maritime traffic. The level of security in AIS has been the subject of many cyber security papers, and several attacks against the system have been identified. VDES is a recent addition to the maritime ICT ecosystem, and aims to unload AIS channels by increasing bandwidth and coverage. With added capabilities, VDES can deliver up-to-date data about weather, water currents, environmental hazards, marine life, and shallow waters, aiding the ship's crew to make informed decisions in time-critical situations.

However, this increase in connectivity and automation also exposes the industry to a new pool of cyber security threats. With a connection to the Internet, additional complexity, and more automated processes, attackers could potentially find new ways of attacking ships, service providers, and authorities in ways previously considered unfeasible.

This master's thesis presents an in-depth analysis of future VDES-based services from the perspective of a malicious actor and how the recent technological advancements will impact the overall security landscape in the maritime industry. We also present findings from two experiments into both AIS and VDES message exchange. In the first experiment, an AIS frame builder was created from scratch in Python to assign spoofed sensor values at runtime. In the second experiment, a VDES frame builder was created from scratch in Python for Gnu Radio Companion, based on the current implementation recommendations. The experiments' results and technical implementations are used to analyze the security in a new maritime application from a cost-based perspective. A comprehensive literature study was also conducted, compiling the technological aspects of these systems based on the latest available recommendations and standards. Findings from three interviews are also presented, where the perception of cyber security in the industry was investigated, and what lessons have been learned from AIS.

Sammendrag

En betydelig økning i antall fartøy i verdenshavene og innlandsfarvann har gradvis blitt mer krevende for maritime ICT-systemer. VTSs, maritime tjenesteleverandører og internasjonale myndigheter sliter med å holde tritt med veksten, da systemer som AIS nærmer seg kapasitetsgrensene sine og utfordrer evnen til å overvåke og håndtere maritim trafikk. Nivået av sikkerhet i AIS har vært temaet for mange cybersikkerhetsrapporter, og flere angrep mot systemet er identifisert. VDES er et nylig tillegg til det maritime ICT-økosystemet og har som mål å avlaste AIS-kanalene ved å øke båndbredde og dekning. Med økte funksjonaliteter kan VDES levere oppdaterte data om værforhold, vannstrømmer, miljøfarer, marint liv og grunne farvann, noe som hjelper skipets mannskap med å ta informerte beslutninger i tidskritiske situasjoner.

Imidlertid eksponerer denne økningen i tilkobling og automatisering også industrien for en ny rekke cybersikkerhetstrusler. Med en tilkobling til internett, økt kompleksitet og flere automatiserte prosesser kan angriper potensielt finne nye måter å angripe skip, tjenesteleverandører og myndigheter på, på måter som tidligere ble ansett som umulig.

Denne masteroppgaven presenterer en grundig analyse av fremtidige tjenester basert på VDES sett fra perspektivet til en ondsinnet aktør og hvordan de nylige teknologiske fremskrittene vil påvirke det overordnede sikkerhetslandskapet i den maritime industrien. Vi presenterer også funn fra to eksperimenter med både AIS og utveksling av VDES-meldinger. I det første eksperimentet ble en AIS-rammebygger opprettet fra bunnen av i Python for å tildele forfalskede sensorverdier ved kjøretid. I det andre eksperimentet, ble en VDES-rammebygger opprettet fra bunnen av i Python for Gnu Radio Companion, basert på nåværende implementeringsanbefalinger. Resultatene og de tekniske implementeringene av eksperimentene brukes til å analysere sikkerheten i en ny maritim applikasjon fra et kostnadsperspektiv. Det ble også gjennomført en omfattende litteraturstudie som samlet de teknologiske aspektene ved disse systemene basert på de nyeste tilgjengelige anbefalingene og standardene. Funn fra tre intervjuer blir også presentert, der oppfatningen av cybersikkerhet i industrien ble undersøkt, og hvilken lærdom som er hentet fra AIS.

Preface

This master's thesis was written in the spring of 2023 in partnership with SINTEF Digital. Karin Bernsed supervised the thesis from the Department of Information Security and Communications Technology (IIK), and with help from co-supervisor Guillaume Bour from SINTEF Digital. The thesis also marks the conclusion of our 5-year M.Sc. degree in Communications Technology and Digital Security from the Norwegian University of Science and Technology (NTNU).

We want to thank both Karin Bernsmed and Guillaume Bour for guiding us in the development and for helpful feedback, insightful discussions and answering all our questions along the way. Thanks to Karin Bernsmed for aiding us with connections to key industry individuals we interviewed for our thesis. Thanks to Guillaume Bour for supporting us throughout our experiments and for helping us debug problems that arose. Thanks to Are Hellandsvik at SINTEF Digital for providing us with the necessary equipment. Also, thanks to our interviewees that participated in our interviews. Finally, we also want to thank SINTEF for providing us with office spaces and equipment, which was an important contributing factor to the execution of our experiments.

Christopher C. Løkken & Brage W. Bugten
Trondheim, 12. June 2023

Contents

List of Figures	xi
List of Tables	xiii
List of Acronyms	xv
1 Introduction	1
1.1 Research Questions	2
1.2 Motivation	3
1.3 Limitations	4
1.4 Thesis structure	4
2 Background	7
2.1 Automatic Identification System	7
2.1.1 How does AIS work	8
2.1.2 Vulnerabilities in AIS	9
2.2 VDES	11
2.2.1 AIS in VDES	11
2.2.2 ASM	12
2.2.3 VDE	12
2.2.4 Maritime Service Portfolio in VDES	14
2.2.5 Security Capabilities of VDES	14
2.2.6 Security challenges in VDES	15
2.3 Hydrographic data exchange	16
2.4 Maritime Connectivity Platform	20
2.4.1 Relationship to other maritime standards	23
2.4.2 Navelink	24
2.5 Related work	25
2.5.1 The feasibility of AIS- and GNSS-based attacks within the maritime industry	25
2.5.2 Cybersecurity of Maritime Communication Systems: Spoofing attacks against AIS and DSC	27

2.5.3	TESLA protocol	27
3	Methodology	29
3.1	Relation to research question	29
3.2	Literature Review	30
3.3	Interviews	31
3.4	Experiments	31
3.4.1	AIS experiment	31
3.5	Resource Cost Analysis of AIS and VDES	32
3.5.1	Resource Cost Estimate Model	33
3.5.2	Cyber Kill Chain	33
3.5.3	Resource Level	34
3.5.4	Estimating Cost	35
3.5.5	STRIDE	36
4	Results from interviews	39
4.1	Interview questions	39
4.1.1	Cyber Security practices, and known AIS attacks	39
4.1.2	How will VDES be utilized, and which services will it be used for	40
4.1.3	VDES security improvements	41
4.2	Key takeaways	41
5	Results from AIS Experiment	43
5.1	Introduction	43
5.2	Motivation and assumptions	43
5.3	Equipment	44
5.3.1	USRP B200mini	46
5.3.2	HackRF One	46
5.3.3	GNU radio	46
5.3.4	AIS_TX	47
5.3.5	gr-ais	47
5.3.6	gr-osmosdr	48
5.3.7	OpenCPN	48
5.3.8	Socat	48
5.4	Set up Transmitter	49
5.5	Set up Receiver	50
5.6	AIS_TX Implementation	51
5.6.1	Frame creation	53
5.6.2	AIS_TX as GRC block	55
5.7	Results	56
5.7.1	Main benefits	57

6	Results from VDE-TER Experiment	59
6.1	VDE-TER Frame builder	59
6.1.1	Generate a VDE-TER message	60
6.1.2	Bit Stuffing	61
6.1.3	Compute and apply CRC32	61
6.1.4	Forward Error Correction	61
6.1.5	Bit Scrambling	65
6.1.6	Apply Syncword and Link ID	65
6.1.7	Byte packing	66
6.1.8	Modulation	66
6.1.9	Results	68
6.2	Building a GNU Radio Signal Simulator for lower leg VDE-TER channel 1024, with Link ID 11	74
7	Analysis	79
7.1	Analysis AIS	79
7.1.1	AIS: Scenario 1, A PKI is available but optional	80
7.1.2	Scenario 2: (CPA) with mandatory signature	87
7.1.3	Scenario 3: Retransmitting AIS position reports over VDE-SAT	91
7.2	Analysis VDES	94
7.2.1	Assumptions	95
7.2.2	Product Description: Under Keel Clearance	95
7.2.3	Description of possible attacks	96
7.2.4	Attack: Exchange set forgery	97
7.2.5	Attack: Manipulate ETA	107
7.2.6	Attack: Update denial	111
7.2.7	Attack: Exchange set replay	114
7.3	Results from AIS and VDES analysis	118
7.4	Challenges of the RCM	121
8	Discussion	125
8.1	Research questions	125
8.1.1	Q1: How will VDES change the security of maritime shipping industry compared with AIS?	125
8.1.2	Q2: How will VDES impact the difficulty associated with attacking maritime communication services?	126
8.1.3	Q3: Will introducing VDES introduce new cyber threats in the maritime industry?	128
8.2	Ethics	129
8.3	Future Work	129
9	Conclusion	131

List of Figures

2.1	Ghost ship is introduced to alter the course of the target (adapted from [GK19])	10
2.2	VDES overview, adapted from [IAL22]	13
2.3	Layer model for s-100 data, adapted from [IAL20b]	21
2.4	MMS Architecture, adapted from [IAL22]	23
2.5	MCP stack, adapted from [IAL22]	24
2.6	Overview Navelink, adapted from [Nav23b]	26
3.1	Relation between research questions and methodologies	30
3.2	Resource Cost Model sub tree, adapted from [Hag20; LB22]	33
3.3	Resource alternatives, and associated colour	35
5.1	USRP and HackRF One used in the experiment	46
5.2	Original AIS_TX setup by [BPW14]	47
5.3	AIS_TX setup with our own modifications [BPW14]	48
5.4	Socat Output	51
5.5	Set up OpenCPN connection	52
5.6	Cyclic Redundancy Check (CRC) lookup table for 16-bit polynomial	54
5.7	Output order of each byte, adapted from [ITU14]	55
5.8	AIS Experiment setup	56
5.9	AIS TX control panel	57
5.10	OpenCPN output	58
6.1	A VDE frame builder (adapted from [ITU22])	60
6.2	General encoder structure, (adapted from [ITU22])	62
6.3	Recursive Systematic Convolutional (RSC) encoder, (adapted from [ITU22])	63
6.4	alternating $\pi/4 - QPSK$ mapping, (inspired by [ITU22])	67
6.5	Append Padding is working correctly	68
6.6	CRC computed correctly	69
6.7	Correct FEC encoding	70
6.8	Result bit scrambling	71
6.9	Extending the modulation scheme	72
6.10	A signal simulator for upper leg channel 1024, with Link ID 11	75

6.11	Results from Frequency Sink and Time Sink for short data messages . . .	77
7.1	Reconnaissance stage, adapted from [WH20]	82
7.2	Weaponization stage, AIS: Scenario 1, adapted from [WH20]	83
7.3	Delivery stage, AIS: Scenario 1, adapted from [WH20]	84
7.4	Exploitation stage, AIS: Scenario 1, adapted from [WH20]	85
7.5	Command and Control, AIS: Scenario 1	86
7.6	Actions on Objectives stage, AIS: Scenario 1, adapted from [WH20] . . .	87
7.7	Weaponization stage, AIS: Scenario 2	89
7.8	Exploitation stage, AIS: Scenario 2	90
7.9	Delivery stage, AIS messages transmitted via Satellite-based VDES (VDE-SAT)	93
7.10	Data set forgery sequence diagram	98
7.11	Screenshot from marinetraffic.com	99
7.12	Reconnaissance stage for UKCM	100
7.13	Weaponization stage for UKCM	101
7.14	Delivery stage for UKCM	102
7.15	Exploitation stage for UKCM	103
7.16	Installation stage for UKCM	104
7.17	Command and control stage for UKCM	105
7.18	Actions on objective stage for UKCM	106
7.19	ETA manipulation attack sequence diagram	107
7.20	Command and Control step for ETA attack	109
7.21	Actions on Objective step for ETA attack	110
7.22	Update denial sequence diagram	111
7.23	Delivery step for Update denial attack	112
7.24	Command & control step for Update denial attack	113
7.25	Replay attack sequence diagram	115

List of Tables

2.1	Position report, (adapted from [ITU14; LB22])	8
2.2	Maritime Service Portfolios	14
2.3	S-100 product specification, adapted from [IHO23a]	18
2.4	RCM results, Closest Point of Approach (CPA) attack from Walde and Hanus [WH20]	27
3.1	Overview of security properties violated by STRIDE (adapted from [Poo20])	36
5.1	Values that we chose to focus on for our experiment	53
6.1	Short data message with acknowledgment, adapted from [ITU22])	60
6.2	Link IDs and their associated parameters for interleaving, (adapted from [ITU22])	63
6.3	Puncturing pattern for some Link IDs , (adapted from [ITU22])	64
6.4	Puncturing pattern for tail bit periods , (adapted from [ITU22])	65
6.5	Link ID code words, (adapted from [ITU22])	66
6.6	Modulation schemes for different Link IDs, (adapted from [ITU22]) . . .	66
6.7	Correct symbol mapping from ITU, [ITU22]	72
6.8	Result of our Symbol mapping	73
6.9	Protocol Format for 6-bit ASCII text, adapted from [IAL22])	75
7.1	Protocol format of signed AIS message (adapted from [IAL22])	80
7.2	Estimation of Cost	87
7.3	Total Estimation of Cost	91
7.4	Retransmit AIS messages over VDE-SAT (adapted from [IAL22])	92
7.5	Estimated cost for Scenario: 3	94
7.6	Total cost of Data forgery attack	106
7.7	Total cost of ETA attack	110
7.8	Total cost of Update denial attack	114
7.9	Total cost of Data replay attack	116
7.10	Estimated total cost results	118

List of Acronyms

- AES** Advanced Encryption Standard.
AIS Automatic Identification System.
AIVDM Automatic Identification System Vessel Data Message.
API Application Programming Interface.
ASC Announcement Signaling Channel.
ASM Application Specific Messages.
AtON Aids to Navigation.
- BB** Bulletin Board.
- CA** Certificate Authority.
CBC Cipher Block Chaining.
CD Compact Disk.
CKC Cyber Kill Chain.
CLI Command Line Interface.
CMDS Common Maritime Data Structure.
COG Course Over Ground.
COTS commercial off-the-shelf.
CPA Closest Point of Approach.
CRC Cyclic Redundancy Check.
CRL Certificate Revocation List.
- DDoS** Distributed Denial of Service.
DoS Denial-of-Service.
DSA Digital Signature Algorithm.
DSS Digital Signature Standard.
- ECDIS** Electronic Chart Display and Information System.
ECDSA Elliptic Curve Digital Signature Algorithm.
ENC Electronic Navigational Chart.
- ETA** Estimated Time of Arrival.
EU European Union.
FEC Forward Error Correction.
GMSK Gaussian Minimum Shift Keying.
GNSS Global Navigation Satellite System.
gr-ais GNU Radio Automatic Identification System.
GRC GNU Radio Companion.
GT Gross Tonnage.
GUI Graphical User Interface.
HW_ID Hardware ID.
- IALA** International Association of Marine Aids to Navigation and Lighthouse Authorities.
ICT Information and Communication Technology.
IDS Intrusion Detection System.
IHO International Hydrographic Organization.
IMO International Maritime Organization.
IP Internet Protocol.
ITU International Telecommunications Union.
JWT JSON Web Token.
M2M Machine to Machine.

MAS Maritime Assistance Service.
MCP Maritime Connectivity Platform.
M_ID Manufacturer ID.
MIPS Million Instructions Per Second.
MIR Maritime Identity Register.
MiTM Man-in-The-Middle.
M_KEY Manufacturer Key.
MMS Maritime Messaging Service.
MMSI Maritime Mobile Service Identity.
MMTP Maritime Messaging Transport Protocol.
MRN Maritime Resource Name.
MSB Most Significant Bit.
MSC Maritime Safety Committee.
MSI Maritime Safety Information.
MSP Maritime Service Portfolio.
MSR Maritime Service Registry.

NIST National Institute of Standards and Technology.
NPIO Nautical Publication Information Overlay.
NRZI Non Return Zero Inverted.

OCC Opportunity Cost of Capital.
OpenCPN Open Chart Plotter Navigator.
OSCP Online Certificate Status Protocol.
OSI Open System Interconnection.

PKI Public Key Infrastructure.

RAC Random Access Channel.
RAIM Receiver Autonomous Integrity Monitoring.
RCM Resource Cost Model.
REST Representational State Transfer.
RSC Recursive Systematic Convolutional.

SA System Administrator.
SAR Search and Rescue.
SART Search and Rescue Transponder.
SATCOM Satellite Communication.
SBB Satellite Bulletin Board.
SDR Software Defined Radios.
SHA Secure Hash Algorithm.
SIP Strategy Implementation Plan.
SMMP Secure Maritime Messaging Protocol.
SOG Speed over Ground.
SOLAS Safety of Life at Sea.
SOTDMA Self-Organizing Time Division Multiple-Access.

TBB Terrestrial Bulletin Board.
TC Threat Concequence.
TCP Transmission Control Protocol.
TDMA Time Division Multiple-Access.
TESLA Timed Efficient Stream Loss-tolerant Authentication.

UKCM Under Keel Clearance Management.
UN United Nations.
URN Uniform Resource Names.
USB Universal Serial Bus.
USRP Universal Software Radio Peripheral.

VDE VHF Data Exchange.
VDES VHF Data Exchange System.
VDE-SAT Satellite-based VDES.
VDE-TER Terrestrial VDES.
VDL Very High Frequency Data Link.
VHF Very High Frequency.
VIS Voyage Information Service.
VPFI VDE Protocol Format Identifier.
VTS Vessel Traffic Services.

Chapter 1

Introduction

Trading has been a source of wealth and power and an essential part of human history, as a critical driver behind cultural evolution, globalization, and economic growth. Archaeological findings dating as far back as 2000 BCE [HIB+07] tell a story of how the indigenous people of Taiwan processed and traded jade with other island communities in the Philippines. Still, to this day, we find new ways of optimizing trade routes through technological advancements, saving time and resources to make the journey as short and efficient as possible. Today, it is estimated that maritime routes carry about 90% of all global trade, and even amongst European Union (EU) countries, shipping accounts for up to 40% [EUS16; LB22]. Shipping is crucial in the global economy, and its contribution has only increased rapidly since the cold war.

According to Aziz et al. [ATSP20], more than 650000 vessels are currently estimated to operate in the world's oceans, transporting commodities, capital, and services through international waters and borders. ICT systems have consequently gained more importance, making up a crucial part of day-to-day operations for personnel on both water and land. International organizations such as IMO and International Association of Marine Aids to Navigation and Lighthouse Authorities (IALA) help mediate international cooperation through regulations, standardization, governance, and harmonization of new technology to meet rising demands with an increased focus on security, availability, capacity, and safety at sea.

AIS is one of these ICT systems that were introduced to increase maritime visibility by periodically sending out vessel reports to nearby ships and shore-based ground stations. Essential information about the ship, such as position, speed, heading, and ship identity, are made available to all receivers within a 70 km radius [ESA23], further enabling a range of maritime applications such as collision avoidance, fleet monitoring, SAR operations and surveillance. Since the Safety of Life at Sea (SOLAS) convention in 2002, every ship above 300 Gross Tonnage (GT) is required to carry a AIS transmitter on board [Alé21].

Today, AIS is also used to carry Application Specific Messages (ASM), which provides ships with even more contextual information such as local weather information, oil spills, shallow waters, marine life, ship berth information, number of persons on board report, SAR search patterns, water level and much more. AIS was not initially intended to scale this way, and channels are under heavy load, especially in high-traffic areas such as the largest ports. Additionally, AIS has been subject to scrutiny by many cyber security papers, as no authentication or confidentiality is provided [BPW14; ATSP20; Lit21; WH20].

Capacity, availability, and security are some of the main challenges with modern AIS. A widely considered successor for AIS is VDES, which aims to solve many challenges by introducing additional channels and half and full duplex communication via satellite and ground stations. With these fundamental improvements, new maritime services, which previously were impossible to provide, are now being implemented, and new standards are already being developed to meet the needs of future maritime applications. Our thesis investigates how this transition to VDES impacts security in maritime applications and what possible new threats are introduced due to ships becoming more connected. The analysis is based on three main research questions the thesis attempts to answer.

1.1 Research Questions

The security challenges of AIS is a well-researched area. However, since VDES is a new technology currently under testing, limited research exists on the possible security challenges associated with VDES. In this master's thesis, we wish to explore the potential cyber security challenges in VDES and have therefore identified the following research questions maintained from the pre-project [LB22].

- RQ-1: How will VDES change the security of maritime shipping industry compared with AIS?
- RQ-2: How will VDES impact the difficulty associated with attacking maritime communication services?
- RQ-3: Will the introduction of VDES introduce new cyber threats in the maritime industry?

A cyber threat is defined by NIST as:

“Any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation),

organizational assets, individuals, other organizations, or the Nation through an information system via unauthorized access, destruction, disclosure, or modification of information, and/or denial of service.” [NIS23]

In our case, the information systems in question are maritime services, and the organizations are the industry stakeholders, such as the shipping companies and maritime service providers.

1.2 Motivation

The Maritime Safety Committee has since 2006 been leading the coordinated work of establishing an international E-Navigation strategy. The IMO defines E-Navigation as:

“The harmonization, integration, exchange, presentation and analysis of marine information on board and ashore by electronic means to enhance berth-to-berth navigation and related services for safety and security at sea and protection of the marine environment.” [IMOa]

The main goal of the E-Navigation strategy is to harmonize existing maritime navigation systems and improve digital infrastructure and information. VDES is an integral part of the implementation strategy for E-Navigation, and the IMO envisions VDES as a possible service delivery method for services such as Maritime Safety Information [IMOC]. Some of the identified benefits of this strategy are [IMOb]:

1. Improved safety at sea by supporting decision-making from improved information and consistent coverage as well as system availability and resilience while reducing human errors by employing automatic procedures, warnings and fail-safes.
2. Supporting the effort towards protecting the environment against pollution and spills by reducing the chance of collisions while enhancing the ability to handle and respond to emergencies.
3. Reducing costs associated with operations by standardization of equipment, automated reporting, and reducing the administrative burden, which will enable relocation of resources to more critical systems.

The rate of maritime accidents is still high [Voy22], which is one of the main motivations behind the E-navigation strategy. VDES has an important role to play in saving lives and improving safety for both humans and the environment. By increasing coverage and reliability, VDES lays the foundation for a range of new maritime services which supports seafarers with different levels of automation and

produce essential data in time-critical situations. Less overhead translates to more time for decision-making for the people on board, which is a key benefit of automation. Adopters of this new technology also need to assign a level of trust and decide the weight of the information they provide in critical situations. To ensure that the E-navigation strategy reaches its goals, it is essential to protect these systems against malicious actors. By analyzing the security of future VDES-based services, end users can be better suited to assign an appropriate level of trust to the communication systems they rely on. This is the primary motivation behind our master's thesis.

1.3 Limitations

In our pursuit to answer our research questions, the thesis aims to investigate the security capabilities of VDES, how this improves the current situation, and what new vulnerabilities and threats are introduced. While we attempt to draw a full-fledged picture of the security and threat landscape, our research is limited to the material at the time of writing. Currently, VDES is still in its early stages of development, and mainly proposals and recommendations are available. This uncertainty means that some technical details could change in the final version as test beds evolve and hidden issues are resolved. Our analysis is therefore based on some assumptions about what will be included, and we have attempted to answer our research questions in this context. Therefore it was necessary to write two new tools from scratch for the practical experiments. The assumptions we make may or may not hold in the long run. However, we still consider the analysis to be of great value, as we can set a lower bound, given more conservative assumptions. We also believe this will be of value to stakeholders in the maritime industry, as they may more easily conduct a cost-benefit analysis of their own based on our results.

1.4 Thesis structure

Firstly, we present the background work in Chapter 2, which goes into the technical details of AIS and VDES, as well as related work. Since VDES is still in development, the background also presents how VDES is envisioned to be used in conjunction with other components in the ecosystem to deliver services to consumers. We then shift to presenting the main work, and we start this part by explaining the various methodologies used to carry out the research in Chapter 3. We then present our interview findings section in Chapter 4, where we gained valuable insight from three stakeholders in the industry. Building on the background and interviews, we present our setup and results from an experiment conducted on the security of AIS and an experiment on Terrestrial VDES (VDE-TER) in Chapter 5 and Chapter 6. The experiments give a perspective on the security impact of open-source software on these systems in light of recent technological advancements, which we further explore

in our analysis. The analysis in Chapter 7 presents a cost estimate for carrying out attacks against AIS and VDES applications. The results from both experiments and analysis are further discussed in Chapter 8, where we revisit our research questions in the context of our results. Finally, we highlight the main points of the thesis in the conclusion.

Chapter 2

Background

This chapter lays the foundation for our thesis and presents the main findings from our literature review. These findings were further used to prepare our interviews and devise our experiments and analysis. We first present the technical characteristics of AIS, followed by the proposed technical recommendations for VDES. Note that the sections regarding AIS and VDES are based on the pre-project [LB22]. We then put these technologies into a larger picture and describe how they fit into the maritime ICT ecosystem. We end this chapter by presenting related work that inspired our experiments and analysis. The reasoning behind creating a background is explained in Chapter 3, which goes in-depth on the various methodologies used in this thesis.

2.1 Automatic Identification System

A similar summary was carried out in the project preceding this thesis [LB22]. AIS is an automatic tracking system initially developed as a collision avoidance tool for the maritime industry. Since 2002, AIS has been mandatory for ships over 300 tonnage and all passenger ships, corresponding to over 300 000 vessels worldwide [WH20; BPW14]. In order to identify different stations, each AIS station is identified by an Maritime Mobile Service Identity (MMSI).

To deliver its position, AIS calculates its Speed over Ground (SOG) and Course Over Ground (COG) from Global Navigation Satellite System (GNSS) data [ITU14] before transmitting positional reports over AIS. Depending on the ship's speed and course, positional reports are transmitted over AIS from every second up to every third minute [ITU14].

Today AIS messages are used to avoid collisions, but they are also used for accident investigations, search and rescue operations, and maritime traffic monitoring, to mention a few applications. The range of AIS without repeaters is roughly 40 nautical miles [ESA23]. Stations equipped with AIS receivers can also forward AIS

messages to different internet sites [GK19], thereby providing valuable data for global monitoring sites such as marinetraffic¹.

2.1.1 How does AIS work

AIS messages are transferred over two VHF AIS channels, A and B, operating at 161.975 and 162.025 MHz respectively [BPW14], with a bandwidth of 25 kHz [ITU14]. The access scheme used for positional messages 1 and 2 are Self-Organizing Time Division Multiple-Access (SOTDMA), and all AIS messages are modulated with frequency modulated Gaussian Minimum Shift Keying (GMSK) [ITU14]. The technical specifications for AIS contain 27 defined message types. In Table 2.1, message types 1, 2, and 3 are included.

For parameters containing text, 6-bit ASCII encoding is applied [LB22]. For the resulting bits, a 16-bit CRC check sequence is calculated and appended [ITU14]. Then the training sequence, start flag, end flag, and buffer are appended accordingly. Each byte for the entire frame should be appended with the least significant bit first before Non Return Zero Inverted (NRZI) encoding is performed [ITU14]. Finally, the frame is transmitted according to the AIS access scheme. Frame generation is explained in greater detail in Chapter 5.

Table 2.1: Position report, (adapted from [ITU14; LB22])

Parameter	Number of bits	Description
Message ID	6	Message identifier, 1,2 or 3
Repeat Indicator	2	How many time has the message been repeated
User ID	30	MMSI
Navigational Status	4	Different values for different states
ROTAIS	8	Rate of turn
SOG	10	Speed over ground in 1/10 knot steps
position Accuracy	1	The position accuracy flag, depending on how accurate the report is

Continued on next page

¹<https://www.marinetraffic.com>

Table 2.1: Position report, (adapted from [ITU14; LB22]) (Continued)

Parameter	Number of bits	Description
Longitude	28	Longitude
latitude	27	Latitude
COG	12	Course over ground
True Heading	9	Degrees
Time stamp	6	UTC second of when the AIS report was generated
Maneuver Indicator	2	Is a special maneuver conducted
Spare	3	Reserved for future use
RAIM-flag	1	1 or 0 depending on if RAIM is used
Communication state	19	What Access scheme is used

2.1.2 Vulnerabilities in AIS

It has been known for a long time that AIS is vulnerable to various attacks. This is due to AIS traffic being neither encrypted nor authenticated [GK19; BPW14; WH20]. In 2014, Balduzzi et al. did a security evaluation of AIS [BPW14]. Using Software Defined Radios (SDR)s and open-source Software such as GNU Radio, they showed how a potential adversary could exploit the vulnerabilities in AIS to tamper with the maritime industry. This section aims to summarize their findings.

Ship Spoofing

The first threat identified by Balduzzi et al. was how an adversary could spoof a valid but non-existent ship. To do this, the adversary must assign the parameters shown in table 2.1. An adversary, for example, a pirate, can use this to craft a valid non-existent ship on the collision course of the target. This scenario is depicted in Figure 2.1, where pirates transmit false AIS messages to lure the target into altering its course into dangerous waters.

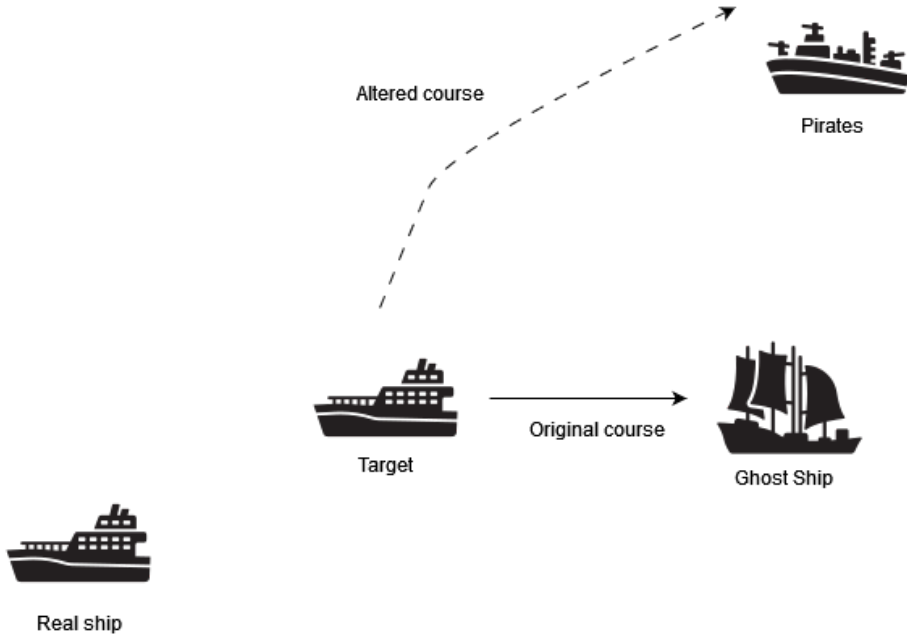


Figure 2.1: Ghost ship is introduced to alter the course of the target (adapted from [GK19])

As AIS is primarily a collision avoidance tool, an adversary could spoof a possible collision. By design, AIS enables automatic response upon detection of a possible collision [BPW14]. This feature is called CPA, and an attacker can utilize this to spoof a ship on a collision course with the target. This could trigger an alert from the targets CPA system, which could lead the ship to enter dangerous waters or result in the ship navigating into shallow ground [BPW14].

Aids to Navigation and Search and Rescue Transponder spoofing

AIS is also used in Aids to Navigation (AtoNs), to assist with traffic management in harbors. An adversary could spoof AtoN messages by placing a malicious buoy in a harbour. Fake information could then be broadcasted, resulting in ships possibly navigating into low water and crashing. Search and Rescue Transponder (SART) also use AIS. When a SART transponder comes in contact with water, the SART device automatically sends out a distress beacon and its GNSS position [BPW14]. An adversary could use this to trigger a fake SART alarm and lure its target into dangerous waters. By law, upon receiving a search and rescue message, the receiver must react [BPW14].

AIS Hijacking

In a AIS hijacking attack, an adversary modifies valid AIS messages by modifying different parameters. An adversary could craft valid AIS messages with wrong data and transmit them using SDRs. If the adversary's target is not the actual ship but an actor relying on correct information regarding the ship. An adversary could forward the malicious data to online providers [BPW14].

Availability Disruption Threats

Three attacks regarding availability disruption were identified by Balduzzi et al., namely *Slot Starvation*, *Frequency Hopping*, and *Timing Attack* [BPW14]. In a slot starvation attack, the adversary denies its victim the ability to transmit AIS messages by impersonating a maritime authority and reserving the entire AIS frame. Similarly, in a frequency hopping attack, the adversary impersonates a maritime authority and asks its victim to change their frequency. By protocol, the receiving station has to change its frequency. In a timing attack, the adversary instructs its target to delay its transmission time. By continuously instructing its target to delay, the target can no longer broadcast its position, resulting in the vessel disappearing from AIS enabled radars [BPW14].

2.2 VDES

Following AIS, VDES is the state-of-the-art radio communication and data exchange system currently being developed. AIS has evolved continuously to accommodate an increasing number of applications, moving away from its original purpose as an identification and collision avoidance system. The motivation behind developing VDES stems from the existing limitations of AIS, which today suffers from saturated communication channels. The proposed solution is to separate AIS messages from ASM by introducing separate AIS and ASM channels, respectively. These channels will provide the same service that AIS does today but reduces a substantial load from existing AIS communication channels. In addition, VDE-TER and VDE-SAT channels are introduced, providing ships with two-way communication using a high-rate data link.

2.2.1 AIS in VDES

AIS continues to be an important component in VDES, as services related to location and identification such as VTS, SAR locating, tracking and Very High Frequency Data Link (VDL) control are provided over AIS in VDES [Bob15]. According to Inter-

national Telecommunications Union (ITU), the technical specification of AIS in VDES continues to be defined by the technical specification in ITU-R M.1371² [ITU22].

2.2.2 ASM

ASM has previously also been provided by AIS, and allows for customized messages for specific applications. The format of messages needs to be initially defined in the IALA ASM collection [IAL11] before the ASM can be utilized. These are often also specific to a region, which causes the number of different ASMs to increase steadily. These messages are assigned dedicated channels in VDES, thereby unloading the AIS channels. With VDES, the capacity of ASM messages that can be transmitted will increase significantly [NB]. The transition of moving all ASMs to VDES ASM channels will happen gradually [Bob15].

2.2.3 VDE

VHF Data Exchange (VDE) is another component of VDES and provides robust half-duplex high-speed data exchange channels. This paves the way for a range of future applications, which have previously not been possible with simplex channels. VDE is further split into two parts, namely VDE-TER and VDE-SAT. These channels enables third-party providers to develop new applications using a “single-window” approach, where a single trusted point of contact collects data related to safety and navigation. Maritime Assistance Service (MAS) or VTS centers can be used for this purpose and are also able to broadcast relevant information in return and provide service to ships within coverage [Waw15].

VDE employs the use of Time Division Multiple-Access (TDMA) and uses six TDMA channels. In practice, this means that every 6th slot belongs to the same channel, and every 15 slots are defined as a TDMA channel frame. Since there are 2250 slots in a minute and $2250/6 = 375$ slots per TDMA channel, there are a total of $375/15 = 25$ TDMA frames per channel each minute which is governed by the Bulletin Board (BB). Here the VDES recommendations differentiate between Terrestrial Bulletin Board (TBB) and Sattelite Bulletin Board (SBB), with the main difference being which slots are fixed for different functions such as signaling, access, and data transmission [ITU22; IAL22]. For instance, the SBB also defines Announcement Signaling Channel (ASC), which the TBB does not have.

The ship needs to adjust its transmission to these changes based on what BB is received. Both SBB and TBB are transmitted at the start of each frame. For transmission, VDE uses 12 different 25 kHz channels in the 157.200 MHz to 161.925 MHz frequency range, intended for both terrestrial and satellite use. The lower six are

²<https://www.itu.int/rec/R-REC-M.1371>

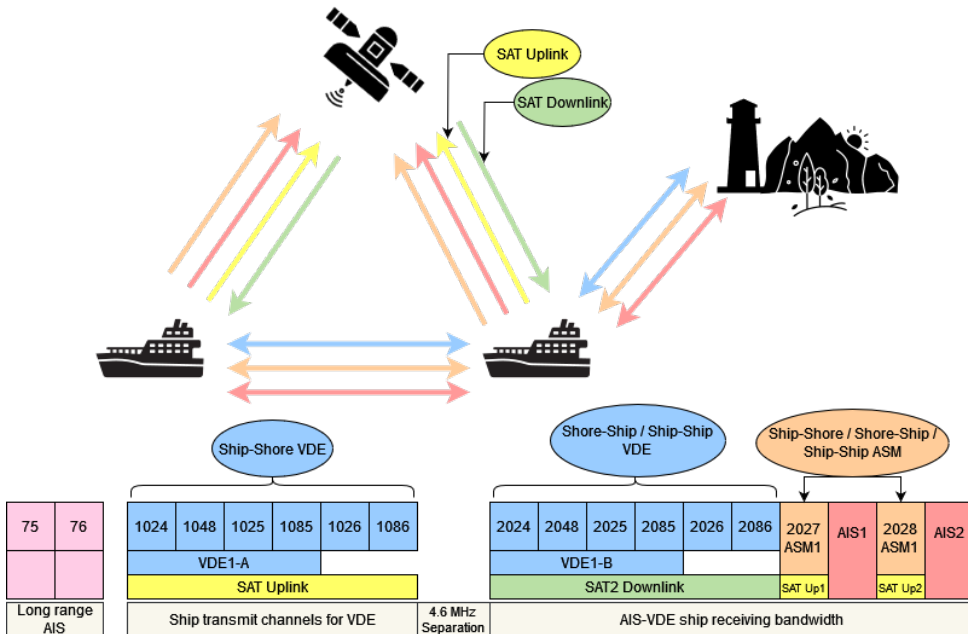


Figure 2.2: VDES overview, adapted from [IAL22]

used for ship transmission, and the upper 6 for ship reception. These two physical channels are separated by a 4.45 MHz guard band, and can be used in 25 kHz, 50 kHz and 100 kHz mode for higher data rates.

Eriksen et al. [EBHS16] noted that raw data rate of up to 307.2 kbit/s is possible, which is 32 times that of AIS which is able to carry 9.6 kbit/s [ITU22]. Finally, VDE channels are also able to transfer arbitrary data [IAL22; LRB], which is in contrast to the fixed message formats found in AIS and ASM.

VDE-TER

VDE-TER refers to the terrestrial part of VDES and has the main goal of providing ship-to-ship, ship-to-shore and shore-to-ship communication. In this system, ships use physical channels 1024, 1084, 1025, and 1085 (also called the “lower leg”) to transmit to shore and other ships and receive from shore stations. Furthermore, ships use physical channels 2024, 2084, 2025, and 2085 (also referred to as the “upper leg”) for receiving from other ships and shore stations [ITU12].

VDE-SAT

VDE-SAT refers to the satellite part of VDES, and has the main goal of providing ship-to-satellite and satellite-to-ship communication. The same physical channels used in VDE-TER might be used with VDE-SAT without imposing constraints on VDE-TER [IAL22]. Channels 1026, 1086, 2026 and 2086 are identified solely for VDE-SAT, not used by VDE-TER [ITU22].

2.2.4 Maritime Service Portfolio in VDES

The IMO and Maritime Safety Committee (MSC) identified 16 Maritime Service Portfolios (MSPs), which are proposed services that supports the e-navigation SIP [Mar18]. The proposed services are listed in Table 2.2 and outline the key areas of improvement contributing to meeting demand and solving the industry’s challenges. In the same report, the MSC also identified VDES as a key enabler of realizing the e-navigation SIP. The development of these services also becomes a driving factor in the development of VDES, as they incentivize end users to make the transition from AIS.

MSP1	VTS Information Service (INS)
MSP2	VTS Navigational Assistance Service (NAS)
MSP3	Traffic Organization Service (TOS)
MSP4	Local Port Service (LPS)
MSP5	Maritime safety information (MSI) service
MSP6	Pilotage service
MSP7	Tug service
MSP8	Vessel shore reporting
MSP9	Telemedical assistance service (TMAS)
MSP10	Maritime assistance service (MAS)
MSP11	Nautical chart service
MSP12	Nautical publications service
MSP13	Ice navigation service
MSP14	Meteorological information service
MSP15	Real-time hydrographic and environmental information services
MSP16	Search and rescue (SAR) service

Table 2.2: Maritime Service Portfolios

2.2.5 Security Capabilities of VDES

The security capabilities of VDES were reviewed in the project preceding this master’s thesis [LB22]. This section is amended with information from the technical

specifications of VDES [ITU22]. AIS suffers from a lack of security mechanisms, which exposes the services and users dependent on the system, as discussed in section Section 2.1.2. Many mitigation techniques have been proposed to solve this problem. However, as strong cryptography often relies on computational power and bandwidth, these are scarce resources that have not yet been prioritized for securing communications. This is one of the main benefits of VDES which increases the overall bandwidth using terrestrial and satellite VDE channels.

To secure communications in VDES, a Public Key Infrastructure (PKI) has been proposed, where a secure communication link can be established between two entities using public and private key pairs and binds the public key to an identity utilizing a digital signature from a trusted Certificate Authority (CA). This is a widespread mechanism already used on the Internet to provide capabilities such as confidentiality, integrity, and authenticity. The most commonly used standard for implementing a PKI is the X.509 standard and is the standard that VDES is planning on using [ITU22]. The cryptographic algorithm intended is the elliptic curve digital signature algorithm with the curve secp256r1, and Secure Hash Algorithm (SHA)-256 as the message digest algorithm. This will offer 128 bits of minimum security [ITU22; PHT+09].

Currently, authentication is not part of AIS. However, authentication is an important feature of VDES, and the use of ASM and VDE can provide this feature [IAL22]. The use of Timed Efficient Stream Loss-tolerant Authentication (TESLA), a broadcast authentication protocol, has also been proposed for this application. TESLA can authenticate broadcast messages using a combination of asymmetric and symmetric cryptography [IAL22]. TESLA is explored in more detail in Section 2.5.3, which includes an explanation for how TESLA manages to achieve a low overhead which is ideal for VDES capabilities.

To avoid replay attacks, some security-critical exemplified VDE messages have a parameter, “valid until” that consists of 32 bits, that is included as a parameter in the signature generation [IAL22].

2.2.6 Security challenges in VDES

Implementing a PKI in VDES has its challenges, as digital certificates must be revoked when they expire. This problem is thoroughly discussed in [BBBM21], where the main challenge is that ships lose connectivity while at sea and cannot retrieve the Certificate Revocation List (CRL). Digital signatures, which rely on certificates, are essential when sharing documents such as ice charts and potentially automatic reporting, including cargo manifests and passenger lists in the future.

The only mention of digital signatures in the technical specification of VDES [ITU22]

states that a 256 Elliptic Curve Digital Signature Algorithm (ECDSA) should be used for digitally signing the BB, in combination with the SHA256 hash algorithm. The output of the ECDSA is 512 bits and might not be feasible given the available VDES network capacity. This depends on the number of slots available to the ship and would be heavily reduced in areas of high traffic.

The bulletin board is broadcasted from control stations and informs ships of how VDES channel resources are allocated, and a fixed section of the bandwidth is reserved for providing updates to the bulletin board. In the technical specification from IALA [IAL19], the bulletin payload only allocated 32 B (256 bit), so the 256 ECDSA signature does not have sufficient space. This has, however, been changed in the 2022 specification from the ITU [ITU22] where 64 B has been allocated. As public keys of 256 bit can be used, this offers a security level of 128 bit, which is the recommended level given by [PHT+09].

2.3 Hydrographic data exchange

The Electronic Chart Display and Information System (ECDIS) is one of the primary sources of information for ship navigators and can visualize a wide range of information, including Electronic Navigational Charts (ENCs), AIS data, radar and Doppler sonar [Kar22]. ENCs enable paper-less navigation at sea by providing charts on the ECDIS, which visualizes a range of hydrographic data for mariners, such as possible dangers, routes, maritime limits, AtoNs, depths, elevations, etc. Today these charts follow the S-57 standard defined by the International Hydrographic Organization (IHO), which allows ships to retrieve charts from any chart provider and display the relevant information on their ECDIS [IHO23c].

To update these charts, Compact Disk (CD) was the most used method for a long time, where the installation and update CDs were received by mail before the voyage [Muk23]. Today, updates are widely available via Satellite Communication (SATCOM) and use the S-63 standard for signing and authenticating the ENC [IHO20]. S-63 uses certificates on the X.509 format, in addition to the Blowfish algorithm for encryption, and the Digital Signature Standard (DSS) for signature generation, which employs the Digital Signature Algorithm (DSA) as well as SHA-1 as input to the signature function [IHO20]. The current corresponding public key has a length of 512 bits. However, this length is no longer recommended by National Institute of Standards and Technology (NIST) [CMRR23; Bar16].

The security of this scheme is based on a random number generated, which in this instance has a length of 160. According to NIST, this offers 80 bits of security and is too weak for today's standard. Rivest et al. [RHAL92] gives an estimate of the strength of the 512-bit DSA key, as 2.1×10^6 Million Instructions Per Second (MIPS)

years, which is a measurement of a computers processing speed. It is equivalent to a processor processing 2.1 trillion instructions per second, running for a year. This approximation is based on the time complexity of the best-known algorithms for solving discrete logs, such as the number field sieve. Rivest et al. [RHAL92] used the approximation in Equation (2.1) to estimate the strength of the DSA 512-bit, which yields: $L(2^{512}) = 6.7 \times 10^{19} \approx 2^{66}$ instructions.

A parallel implementation of the general number field sieve was done by [XYL05], showing that factoring could be done even more efficiently on multiple cores working simultaneously. Today a 4.35 GHz AMD Ryzen Threadripper 3990X 64 core can output 2.36×10^6 MIPS [Chi20], which means that it would be capable of breaking the scheme in about one year, and is commercially available for 5,000 USD (\approx 50,000 NOK) ³.

$$L(p) = e^{\sqrt{\ln p \ln \ln p}} \quad (2.1)$$

Today the S-57 standard is no longer meeting the requirements of end users in this era of maritime digitization [Skj21]. Consequently, work has already started developing a new standard, namely the S-100 series, which provides an update to ENC's (S-101). It simultaneously introduces a range of new product specifications covering modern applications in the maritime domain. The S-100 standard is a framework for developing new product specifications. It serves as an ecosystem of standards that aim to harmonize how future services are designed, ensuring interoperability across services and international borders. To achieve this, the IHO maintains a publicly available registry of registers, which contains schemas and Common Maritime Data Structures (CMDs)⁴.

So far, 16 product specifications have been created and are shown in Table 2.3 [IHO23b; IHO23a]. S-101 is the new standard for ENC's, and new ENC applications are encouraged to refer to S-101 for development, to replace S-57 ultimately. However, S-57 is still an official standard, and ENC distributors will continue to produce S-57-compliant charts for the foreseeable future [Skj21]. The following product specifications describe data structures and hydro-graphic features that could be used as an information layer on top of an ECDIS, also known as a Nautical Publication Information Overlay (NPIO), complementing the ENC's by adding other navigational data such as depth, areas of interest, surface currents, sea routes, etc.

It is also important to mention that even though the current S-63 standard [IHO20] is cryptographically outdated, the new S-100 standard addresses these challenges

³<https://www.amazon.com/AMD-Ryzen-Threadripper-3990X-128-Thread/dp/B0815SBQ9W?th=1>

⁴<https://registry.iho.int/main.do>

and contains longer cryptographic keys [IHO22]. Notably, the Blowfish algorithm for encryption has been exchanged with Advanced Encryption Standard (AES) in Cipher Block Chaining (CBC) mode, supporting 128, 192 and 256-bit versions. Additionally, both the 20-bit Hardware ID (HW_ID) and Manufacturer Key (M_KEY) have been extended to 128 bits (16 bytes). The signing of S-100-based data products is also revised in the new S-100 standard by extending the use of DSA the System Administrator (SA) (IHO) certificates represents a 2048-bit public key, in contrast from the previous 512-bit key. S-100-based data products are signed using 1024-bit keys in the future, doubling the previous standard. These revisions strengthen the system cryptographically, making it harder for an adversary to forge data products.

According to Barker et al., the new 1024-bit DSA key also offers 80 bits of security. NIST disallows this key length starting from 2030 [Bar20]. The 2048-bit key, on the other hand, provides 112 bits of security and is allowed for applying protection beyond 2031. However, DSA is since February 2023 no longer approved as a part of the DSS. Only verification of existing DSA signatures is allowed. The rationale behind removing DSA is that considerable academic research has found feasible attacks against the signing algorithm [FGR13; GGR19].

According to IALA [IAL22], VDES is a promising candidate for future delivery of ENC's, which would enable ships to update their maps during the voyage while achieving higher availability and reliability. To enable secure exchange of S-100, the IEC 63173-2 SECOM standard was developed, which describes how S-100-based products can be delivered from service providers to end users [MCP]. SECOM uses Internet Protocol (IP) for communication, which in turn can be provided over VDES [IHO23d].

Table 2.3: S-100 product specification, adapted from [IHO23a]

ID	Name	Description
S-100	Universal Hydro-graphic Data Model	This is the foundation for developing future maritime standards that specify what they should contain and how they should be constructed and maintained.
S-101	Electronic Navigational Chart	Vector chart that specifies the ENC's format, how they should be developed, encoded, distributed, etc.
S-102	Bathymetric Surface	Data product that describes and visualizes elevations and formations in the sea floor, allowing ships to navigate shallow waters.

Continued on next page

Table 2.3: S-100 product specification, adapted from [IHO23a] (Continued)

ID	Name	Description
S-104	Water Level Information for Surface Navigation	Data products describing water levels at certain time steps and locations, enabling water level forecasts.
S-111	Surface Current Product Specification	Data product that aims at visualizing velocity and direction of water currents at sea level in discrete points.
S-121	Maritime Limits and Boundaries	Data product that provides information on international borders, zones, and other maritime limits, as described by the UN Convention on the Law of the Sea.
S-122	Marine Protected Areas	Vector data set describing geographical areas of interest which require higher environmental protection standards, for instance, concerning fragile ecosystems.
S-123	Marine Radio Services	Serves as a catalog for maritime radio services and describes what services are available and how to contact them.
S-127	Marine Traffic Management	A dataset that describes the availability of VTSs, local pilotage services, and ship reporting systems. Data will also include how to contact and utilize these services.
S-128	Catalogue of Nautical Products	Catalog describing publications and revisions of documents such as paper charts, ENCs, and other S-100-based products. The catalog will contain their coverage, status, and other metadata.
S-129	Under Keel Clearance Management Product Specification	NPIO describing geographical areas where ships can navigate in relation to under keel clearance, enabling mariners to detect dangerous waterways where the ship could collide with underwater formations such as reefs.
S-201	Aids to Navigation (AtoN) Information	Provides a CMDS for the exchange of AtoN information between authorities.

Continued on next page

Table 2.3: S-100 product specification, adapted from [IHO23a] (Continued)

ID	Name	Description
S-240	Differential GNSS Station Almanac	Data format specification describing how station information can be exchanged between authorities.
S-401	Inland ENC Product Specification	ENC for vessels navigating inland waterways, which includes additional information to the standard ENC. IENC can include sailing directions and machine-readable operating schedules.
S-421	Route Plan	Framework for the production of consistent route plans in maritime applications.
S-98	Data Product Interoperability in S-100 Navigation Systems	Specifies the format and semantics of interoperability catalogs containing rules controlling the interaction between multiple S-100 products. It is intended to reduce information overload when multiple S-100-based data products are used simultaneously, which can potentially clutter the ECDIS display.

2.4 Maritime Connectivity Platform

Maritime Connectivity Platform (MCP) is a decentralized platform that intends to facilitate secure and reliable information exchange in the maritime domain. The concept of MCP consists of three major building blocks, namely Maritime Identity Register (MIR), Maritime Service Registry (MSR), and Maritime Messaging Service (MMS) [IAL22]. MIR is a register that enables authenticity between participants. The authenticity is provided using digital certificates associated with a unique Maritime Resource Name (MRN). Therefore every MCP user needs to have MRN. The MRN syntax is based on the Uniform Resource Names (URN) as described in RFC 2141 [IAL20a].

The MCP vet the MRN according to an accepted vetting procedure [IAL22]. Participants can then use MIR as a CA to authenticate the different services and other participants. Since ships often will not have access to the MIR when abroad, the MIR is cacheable to allow secure communication without a direct connection [IAL22]. The MIR can also be updated to revoke invalidated certificates reported stolen or outdated. If a user wants to validate a certificate, it is possible to check if it has been marked revoked. This can either be done by checking the Online Certificate Status Protocol (OSCP) interface for each certificate or by periodically downloading a certificate revocation file from the identity register [MCC20b].

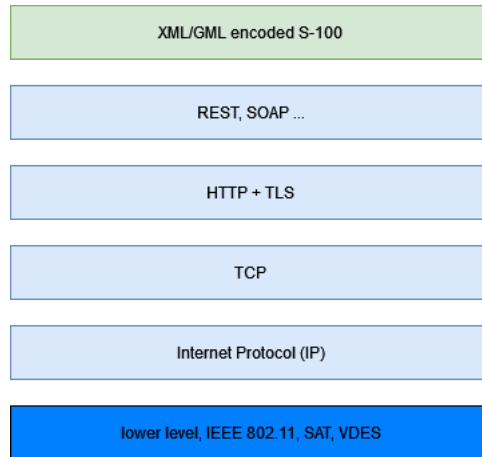


Figure 2.3: Layer model for s-100 data, adapted from [IAL20b]

MMS allows the exchange of digital messages and supports secure IP-based transport through the internet, VDE-TER and VDE-SAT. Figure 2.4 shows the intended MMS architecture. To issue a new certificate for a vessel, an organization administrator needs to log in to a MCP portal and provide information about the vessel, such as MMSI. There are no vetting procedures regarding this information other than the one conducted for the organization [MCC20b]. Figure 2.3 shows the stack intended to transport the actual S-100 data from the service provider to its consumers. In the case of VDES, it can be used as a lower-level communication below IP [IAL20b].

Maritime Service Register

The MSR is a register for service discovery [MCC20d]. The MSR is compared with a yellow phonebook and used as a reference point for finding services [MCC20a]. By design, MSR is intended to be used alongside MIR. To access a service provided in the MSR, the user must authenticate himself using the MIR. In return, the user receives an access token, which it can send to a Service Registry, which returns the service [MCC20d].

The access token is a base 64 encoded JSON Web Token (JWT), passed to the service registry with every request. The organization ID, associated with the users' authentication credentials, is encoded within the token and is set by the Identity registry [MCC20d]. Typically the MSR is accessed using a Representational State Transfer (REST) Application Programming Interface (API) over HTTP. Services might therefore connect to ships over IP, or if IP is not available VDES [MCC20d; IAL22]. To do this, each ship is equipped with an MMS Edge Router that decides

which communication is the best under current connectivity conditions [IAL22].

Maritime Messaging Service

MMS allows the exchange of arbitrary digital messages over secure IP-based transport through the Internet, and through VDE-TER and VDE-SAT [IAL22]. Currently, MMS is still under development, and only draft versions exist online. In the draft, a general description of the MMS architecture is available, consisting of 5 parts:

1. Maritime Messaging Transport Protocol (MMTP): See Section 2.4.1
2. Optional Secure Maritime Messaging Protocol (SMMP): See Section 2.4.1
3. MMS agent: Client software which interfaces with the MMS network using MMTP
4. Edge router: Module which brokers messages originating from or sent to the MMS agents and provides gateway access to the rest of the network.
5. Router network: Zero or more MMS routers, which handle routing and forwarding of MMS traffic to edge routers.

MMS consists of two core functions, message queuing and relaying. Message queuing converts information received from an e-Navigation Service into a message and temporarily stores it. This ensures that even though the communication has deteriorated, the receiver can receive the information stored in the message queue when the communication network is restored. Message relaying is a functionality that enables delivering information to a recipient without a network locator [MCC20c]. In MCP, each user has a unique identifier, an MRN. The source MRN and destination MRN are added to the HTTP message's header. This way, even if the IP address of the user changes, the message can be relayed to the new IP address associated with the MRN [MCC20c].

MMS performs message delivery based on the MRN and forwards the message to the current network locator of the MRN user. Currently, MMS only supports pull operations, pure polling, and long polling [MCC20c]. In pure polling, the client, for example, a vessel, checks if a message has been received at the MMS. If a message has arrived, it is forwarded. If not, an empty message is returned. In long polling, the client sends a polling request to the MMS. The MMS then waits until it receives a message intended for the client is received and then responds. According to the MCP Consortium, the MCP design can also support push. This is not implemented but might be implemented in the future if wanted [MCC20c].

Public Key Infrastructure

The PKI for MCP is a system that creates, manages, distributes, stores, revokes, and uses digital authentication certificates for secure Machine to Machine (M2M)

communication. The PKI infrastructure is based on the X.509 standard, and its key component is the PKI CA [MCC20d]. In the case of MCP, it is envisioned that in the future, each flag state would be its own sub CA that is responsible for issuing certificates for vessels registered under their flag [MCC20d].

MCP Source Code

Interestingly, the MCP consortium maintains its code open-source. This includes the MCP-PKI library for handling certificates⁵, the MIR⁶ API, and the MCP Management Portal, to name a few⁷. The website encourages contributors to contribute and improve the MCP source code [MCC20f].

From a security perspective, this can be seen as both a strength and a potential risk. Having open-source software allows for transparency, making it easier for developers and contributors to review and identify possible vulnerabilities in the code. For a new consortium like MCP, having a large and active community might contribute to the pace of the development and improvement of the software. This also includes identifying security issues that might lead to a more secure platform. However, it might also expose vulnerabilities to an adversary. If vulnerabilities are not addressed fast, they might be used against the MCP before they are patched.

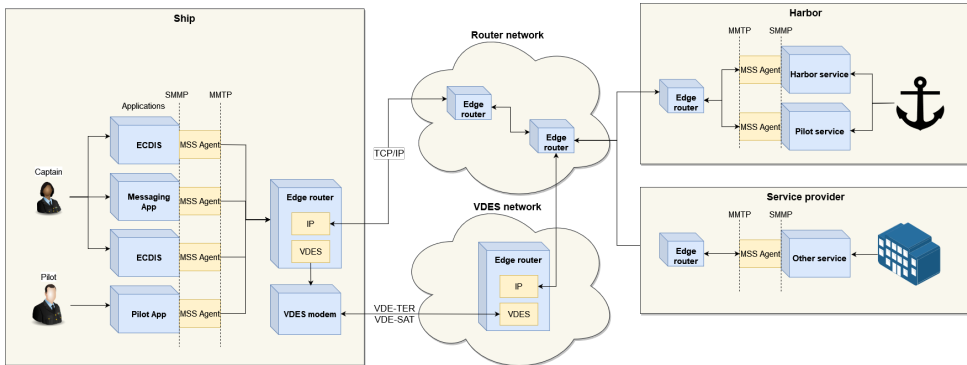


Figure 2.4: MMS Architecture, adapted from [IAL22]

2.4.1 Relationship to other maritime standards

The MCP brings VDES and VDES-based services together by introducing an identity registry for participants and a service registry for discovery. This way, mariners can access S-100 products in the future via MCP using methods described in the IEC

⁵<https://github.com/maritimeconnectivity/MCP-PKI>

⁶<https://github.com/maritimeconnectivity/IdentityRegistry>

⁷<https://github.com/maritimeconnectivity/MCP-Portal>

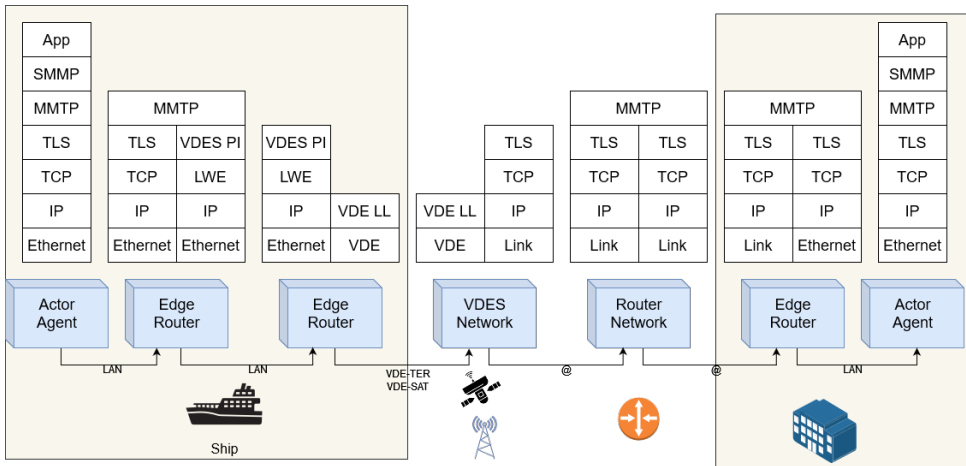


Figure 2.5: MCP stack, adapted from [IAL22]

63173-2 SECOM standard as mentioned in Section 2.3. MCP also introduces the MMTP, which has been created to cover three use cases of the MCP:

1. Registration of agents based on MRNs
2. Transfer of messages between agents
3. Subscriptions to messages

The use of MMTP is shown in Figure 2.5, which illustrates MMTP as a layer in the Open System Interconnection (OSI) layered model. Subscriptions using MMTP allows distribution to multiple MMS agents [IAL22], for instance, in the case of S-100 data products. SMMP runs on top of MMTP, offering end-to-end security in the form of confidentiality, integrity, authenticity, availability, and non-repudiation. However, the protocol is currently optional according to the G1117 [IAL22] draft of SMMP.

S-100-based products are still under development, and it is uncertain what delivery mechanisms make them available to users. The IEC 63173-2 SECOM is one standard that addresses this challenge, as discussed in 2.3. An international standard for the areas not covered by IEC 63173-2 is yet to be published, but the MCP is currently the most promising proposal, which is compliant with both the IEC 63173-2 and VDES

2.4.2 Navelink

An MCP instance provider is an organization that operates a MCP instance [MCC20e]. Organizations are setting up MCP compliant instances for future use. Navelink,

an MCP instance owned by Kongsberg and Wärtsilä, is an example of such an instance [MCC20e].

Navelink currently operates across three environments: development, test, and operational. The company shares its developer forum on its website, allowing external observers to track the development progress of the MCP instance. The most recent developer forum discussion on April 27th, 2023, revolved around enhancing VDES support [Nav23b]. As depicted in Figure 2.6, Navelink envisions a cooperative environment between REST, MMS, and VDES [Nav23b].

However, from our perspective, there appear to be some unresolved issues regarding authentications, service lookup, the service bulletin, and public keys in VDES. During the meeting held on Mars 23rd, 2023, participants discussed how services would be reflected on VDES and whether they would be served behind MMS [Nav23a].

Potential Service Providers must complete a specific process to register a service that can be provided over IP or VDES using Navelink as a MCP. The Voyage Information Service (VIS) Hotel, a server offered by Navelink, is connected to a unique Service identity within the MSR [Nav21]. This design enables ships to retrieve messages whenever they desire. To establish this connection, the customer receives a public URL and links the appropriate client certificate to the service. This configuration ensures that the Service Provider is responsible for the information security within its application. At the same time, Navelink assumes responsibility for the security to and from potential clients [Nav21].

By adhering to this process, Service Providers and Navelink can work together to maintain a secure and reliable maritime communication system, leveraging both IP and VDES through the MCP.

2.5 Related work

2.5.1 The feasibility of AIS- and GNSS-based attacks within the maritime industry

The master's thesis of Walde and Hanus [WH20] looked into the feasibility of attacks against AIS and GNSS where the RCM was used for each attack. In table 2.4, the estimated cost for a AIS-based attack is included. Their thesis inspired us to do the same for VDES and see if the level of feasibility that they observed has changed to any degree since their publication. They concluded that most of the attacks they carried out were feasible. However, the crew's alertness onboard the vessel makes a big difference. Their interviews showed that mariners could navigate without using GNSS and avoid collisions with other ships by observing their surroundings.

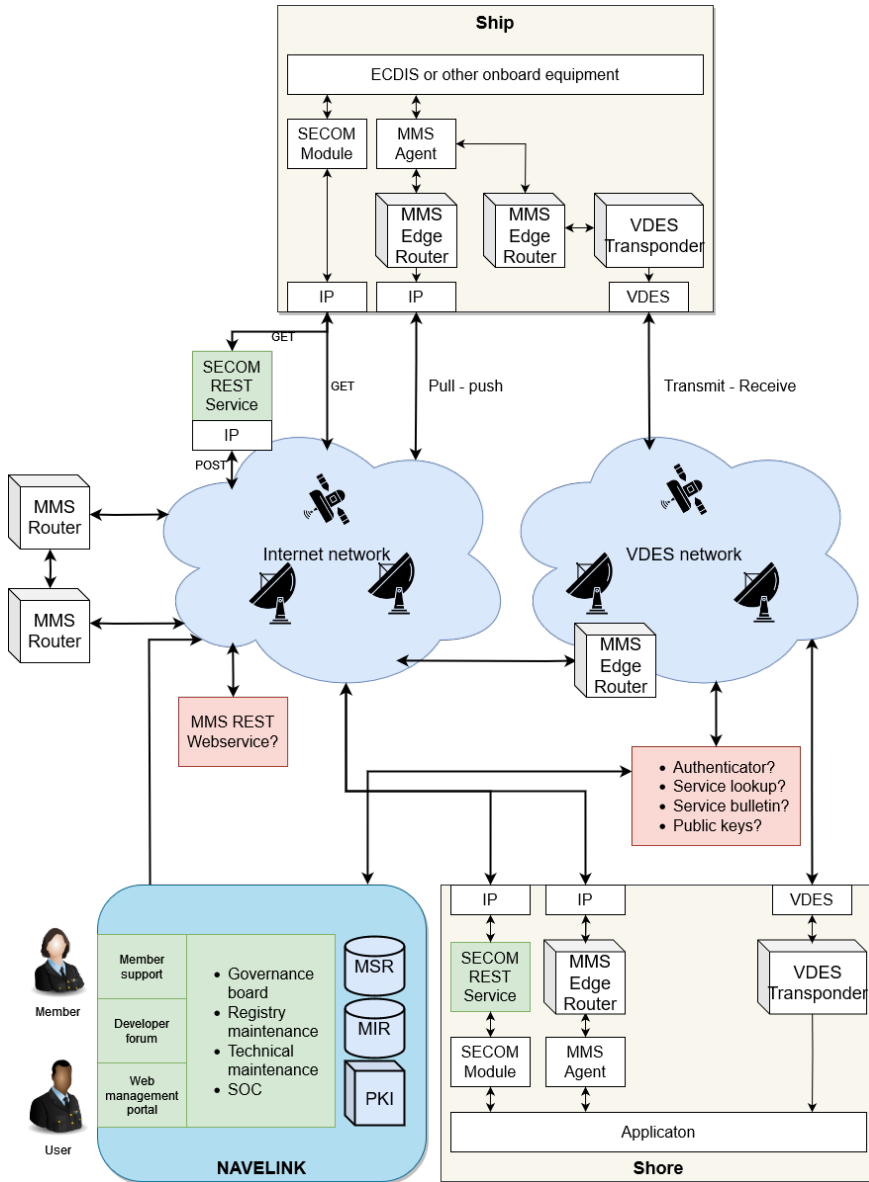


Figure 2.6: Overview Navelink, adapted from [Nav23b]

Their experiments showed that trajectory-based spoofing is possible by choosing appropriate values for course and speed in their AIS position report message.

Table 2.4: RCM results, CPA attack from Walde and Hanus [WH20]

Stage	Min Cost \$ (USD)	Max Cost \$ (USD)	Conf
Reconnaissance	190.80	190.80	1
Weaponization	500	2000	1
Delivery	525	6750	0.1
Exploitation	0	0	1
Installation	625	4750	1
Command & Control	Left out	Left out	Left out
Actions On Objective	0	0	1
Total	1840.8	13690.8	0.1

2.5.2 Cybersecurity of Maritime Communication Systems: Spoofing attacks against AIS and DSC

The master’s thesis of Forsberg [For22] also investigates how susceptible AIS is to spoofing attacks. This was shown through an experiment where some spoofing attacks were carried out. In contrast to Walde and Gaustad, Forsberg instead set up the experiment with the help of a web server that interfaced with the Transmission Control Protocol (TCP) Source block in GNU Radio Companion (GRC). Additionally, the experiment featured a custom web app that was used to create the AIS messages. The messages could furthermore be visualized in the web app. The results of the experiment show that Forsberg managed to conduct spoofing, slot starvation, and hijacking attacks in his setup. However, since Forsberg created his own tools and did not publish the implementation, it is difficult to verify the validity of the results. The setup by Forsberg inspired us to make our own frame builder while using well-known tools such as OpenCPN to validate the results.

2.5.3 TESLA protocol

Litts [Lit21] explains how the TESLA protocol can be implemented for authentication with AIS, which are improvements on the SecureAIS protocol implemented by [ATSP20]. The TESLA protocol can authenticate a stream of packets in a broadcast network by combining asymmetric and symmetric encryption. The sender creates a one-way hash chain of one-time keys and commits the hash of the final key to all receivers in the network, along with a key “reveal schedule” over an authenticated channel, for example, by using a PKI.

For explanation purposes, we can represent the chain of keys as a directed graph

of nodes where each node (except the initial) has exactly one parent and one child, where a child is computed by hashing the parent. The sender must first decide how many keys (nodes) should be computed by evaluating how many packets will be sent. The sender encrypts and broadcasts the first packet by using the first child key, which at first no one can decrypt since only the hash of the required child key has been revealed so far.

After a fixed time period, the child key is revealed, and packets can be decrypted. Because the packet was received before the key was revealed, the sender can be certain that the payload is authentic, as no malicious party could have forged a packet before the key was revealed. Furthermore, the receiver can verify the authenticity of the key by hashing the revealed child key. This is equivalent to moving one step down in the hash chain, and the commit should be a child descendant of the revealed key.

If keys are lost in transmission, the packet sent in advance cannot be verified at first. However, the receiver can compute it by waiting for the lost key's parent and retrieve it by hashing downwards in the chain. The security condition of TESLA is that the receiver can unambiguously decide that the packet was received before the corresponding key has been revealed. The receiver can look at the key reveal schedule and compare it with the timestamp of when the packet was received, but the clocks of the sender and receiver must be loosely synchronized for this to work. This is done with a synchronization request, where the receiver can compute an upper bound for the sender's local clock [Suw16].

Finally, Litts [Lit21] shows how the TESLA scheme performs compared to other methods, such as SecureAIS and X.509 certificates in terms of the number of AIS time slot needed to transmit an authenticated message. From the result of the analysis, TESLA needs between 1.5 to 2 consecutive TDMA frames, SecureAIS needs three or more TDMA frames, while X.509 certificates need 85 frames. A demonstration of TESLA protocol for a AIS-based SDR was implemented in a project by [AA21], using GRC and the AIS Frame Builder from [BPW14].

Chapter 3

Methodology

In order to answer our research questions, we need to apply well-known and studied methodologies which we can rely on to produce the results we are looking for. This section explains our reasoning for choosing the specific methodologies and how they helped us answer the research questions. The methodology has largely remained the same as initially described in the pre-project [LB22]. Since the pre-project, the two experiments have been greatly modified, and the STRIDE threat modeling framework has been added to our list of methodologies.

3.1 Relation to research question

The first research question asks how VDES is intended to be used and what technical design choices have been made by the developers of the standards and recommendations. Hence, we required methodologies that enabled us to investigate the literature as well as the reasoning behind the design choices. It is, therefore, appropriate to employ a Literature Review and further supplement the literature with interviews with industry stakeholders. The literature review lays the foundation for our thesis, and parts of the literature that are unclear were subsequently used as the foundation for the interview questions.

The second research question asks what resources are required to carry out successful attacks against stakeholders in the industry. Additionally, we want to investigate if the required resources change, in terms of cost and availability, with the introduction of VDES as a data exchange mechanism. Hence, RCM was identified as an appropriate methodology as it has been used in earlier papers to analyze AIS, enabling us to compare our results. To verify that an attack is indeed possible and that the estimated cost is well-founded, we also require to simulate the attacks, which we do by conducting two experiments. Finally, the experiments' results help us better understand the actual costs used in the analysis.

The third research question asks what new threats, if any, are introduced to the

industry by VDES. Similar to the second research question, the difficulty of carrying out the attack limits the types of attackers that will target the industry. The results from the experiments and analysis also helped answer this question. Additionally, skill, opportunity, and attacking tools might be more prevalent in other domains, so the type of attackers depends on the system architecture and the underlying technologies. Hence, the technical design choices is a significant contributing factor to what new threats are introduced.

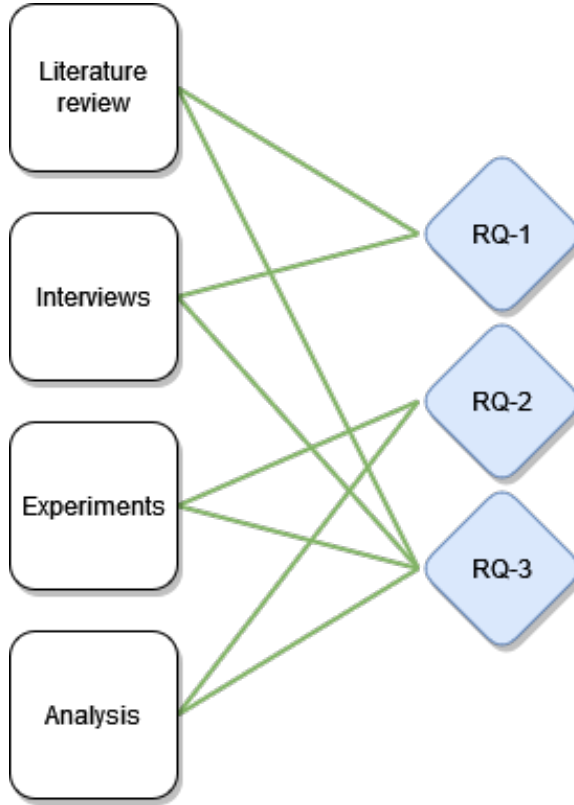


Figure 3.1: Relation between research questions and methodologies

3.2 Literature Review

To provide an answer to the research questions formulated in Chapter 1, a thorough literature review was needed to gain a deeper understanding of VDES. A systematic search was therefore conducted using Google Scholar and Oria as our main search engines to locate peer-reviewed research regarding AIS. However, since VDES is a new concept still under development, research on VDES is very limited. It was therefore deemed necessary to obtain information by looking at publications from

the standardizing bodies. ITU is the UN specialized agency for communication technologies and is responsible for developing the technical characteristics for VDES. Moreover, IALA, an international technical association, is involved in developing VDES and provides guidelines and an overview of VDES.

By using a Snowball Strategy described by [MPM16], we discovered new Standards and technology VDES intends to use. A Snowball strategy is done by examining sources and, in our case, standards and new practices found in papers and then recursively repeating the process. In our case, the snowball effect enabled us to find standards such as the MCP and the new Universal Hydrographic Data Model S-100. Moreover, by visiting the home pages of commercial actors helped provide helpful insights into the latest technology, such as the MCP instance: Navelink. This approach allowed us to find trusted research that, when reviewed, was the latest standard available. Even though a literature review alone is not enough to answer our research questions, it helped lay a foundation for us when assessing the security capabilities of VDES. The results from the literature review were used to write the Background chapter in Chapter 2.

3.3 Interviews

To gain valuable insight into how VDES compares with AIS, its functionality, and possibilities, interviews were conducted with three people associated with the development of VDES. The interviews were done in a semi-structured format that relies on asking questions within a predetermined thematic framework. The interviews were held anonymously, and no participant information was stored. The themes we explored in our interviews were unchanged from the project [LB22] and are the following.

1. What is the general perception of the cyber security level in the maritime industry? Are you aware of the existing vulnerabilities in AIS and GNSS?
2. How do you expect VDES will be utilized in the maritime shipping industry? Which services do you think will be used over VDES?
3. Do you think the introduction of VDES will improve the cyber security within the maritime shipping industry? If yes, how and why?

3.4 Experiments

3.4.1 AIS experiment

Initially, inspired by Balduzzi et al., and Walde and Hanus [BPW14; WH20], we wanted to conduct AIS spoofing attacks using the same setup elaborated on in [WH20] to evaluate the cost of conducting AIS attacks. However, it has been nine years since

the original AIS_TX frame builder was built by Balduzzi et al. [BPW14]. This is a SDR developed to generate and transmit arbitrary AIS packets and was used by the authors to simulate attacks. As Walde et al. [WH20] discovered in their thesis, a lack of documentation and software maintenance lead to various obstacles when building the software. As an example, the original AIS_TX was built for GNU Radio 3.7, and the newest version of GNU Radio is 3.10. We, therefore, decided to research if it was possible to conduct AIS attacks in a more modern context.

As it is illegal to transmit AIS messages in uncontrolled environments [LB22; WH20], a wired connection was used to transmit the signal to ensure the experiment was conducted in a controlled environment. In our AIS experiment, we wanted to see if we could replicate the functionality of the original AIS_TX using an embedded Python block, as this is new in In GNU Radio ≥ 3.8 . We believed this would enable greater room for customization and lower the cost of conducting AIS spoofing attacks.

VDE-TER Experiment

As we could mimic the same results with our embedded Python block, we extended the experiment to see if we could implement a simplified VDE-TER transmitter. Using the same architecture defined by Balduzzi et al. but changing the specific blocks and parameters. Because VDES implementations are currently proprietary, the cost of building a VDE-TER transmitter from scratch was of interest to our analysis. By trying to do this ourselves, we got a more realistic indicator as to the difficulty of setting up and carrying out an attack. The VDE-TER frame builder was built using the technical specification for VDES [ITU22]. Without any open-source solutions regarding VDES available, our results provide substantial aid in future research regarding VDES by contributing with the first available open-source code regarding VDE-TER.

3.5 Resource Cost Analysis of AIS and VDES

As stated at the start of this chapter, we used the RCM to conduct our analysis of AIS and VDES. However, the RCM is not suitable for identifying possible attacks against the systems. To cover this, we used STRIDE, which is a well-known threat modeling methodology. Regarding AIS, only the cost of CPA attacks was analyzed because we want to compare our results with the results obtained by Walde and Hanus [WH20].

From the Literature review and interviews, three different scenarios for CPA attacks were discovered. These are elaborated on in Chapter 7. Therefore, the STRIDE methodology was not required for the AIS analysis. No previous analysis

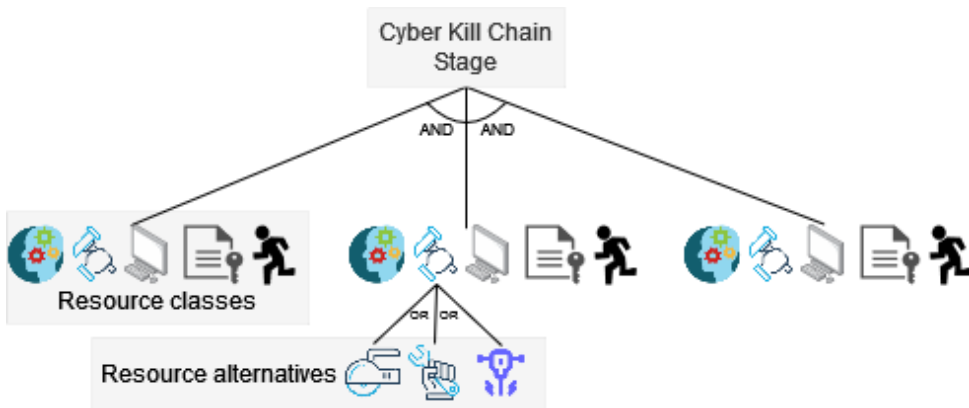


Figure 3.2: Resource Cost Model sub tree, adapted from [Hag20; LB22]

has been conducted for the VDES analysis, so in this case, STRIDE was essential in identifying relevant threats. In this section, each component of the RCM and STRIDE is explained in depth.

3.5.1 Resource Cost Estimate Model

The RCM estimates the total cost of the resources required to carry out a cyber attack in dollars and cents [Hag20] by employing the CKC to define the necessary steps. In the RCM, each stage is coupled with a Resource Tree, which is further divided into three levels: the kill chain step, the resource, and the resource alternatives, see Figure 3.2. For each level, the cost of the required resources is derived.

The resource level contains the resources needed to complete the kill chain, and here one or more resource alternatives can be defined. Resource alternatives are different alternatives that can substitute each other, that the attacker can use to acquire the resource. The essence of the model is that for an attacker to carry out an attack, all stages defined in Section 3.5.2 must be completed [Hag20]. To mitigate a specific attack, it is enough that only one step in the chain is unfeasible for the attacker. For instance, high costs or lack of resource alternatives are amongst the reasons that can make a step unfeasible.

3.5.2 Cyber Kill Chain

The CKC is a framework developed by Lockheed Martin, which identifies what the adversaries must complete to achieve their objective [Loc23]. The methodology describes the stages of a cyber attack, from initial reconnaissance to the successful exploitation of a target, and consists of the following seven stages:

1. Reconnaissance: The adversaries gather information about the target. This might include harvesting email addresses, exploring potential vulnerabilities, and exploring the architecture.
2. Weaponization: The attacker creates the weapon, which might be malware or an exploit, and creates a deliverable payload containing the weapon.
3. Delivery: In the delivery phase, the weaponized bundle is delivered to the victim. This can be done via email, web, USB, or in our case, VDES.
4. Exploitation: The attacker exploits a vulnerability discovered in the reconnaissance phase to execute code on the victim's system.
5. Installation: The attacker installs the weapon, for example, malware, and gets access to the system.
6. Command and Control: The attacker might connect with the compromised system after installing the hardware. Thus allowing the attacker to control it and execute other actions remotely.
7. Action on Objectives: The attacker is ready to achieve their final goal. This might include data ex-filtration, system disruption, or other malicious activities.

3.5.3 Resource Level

The resource level states which resources are required to complete the kill chain. Haga states that in RCM, a resource is classified into five resource classes [Hag20].

1. Skill: Resources classified as skill refer to the ability to develop malware or software necessary to conduct attacks. We have used the same cost model for the resource class skill to compare our results with Walde and Hanus. Namely, that one hour of work costs the adversary \$20, and the total cost is the number of hours worked times the cost per hour [WH20].
2. Tangible: Tangible resources are hardware components or other physical objects required to conduct the attack. For example, the hardware might vary from a computer to an SDR. Such items are easy to derive the cost for as they have a specific price associated with them.
3. Logic: Logic resources are commercially available software and tools. Examples are malware which can be bought online and GitHub repositories.
4. Logic-atomic: Logic-atomic resources can't be broken into lesser resources. Examples include emails, passwords, and private and public keys.

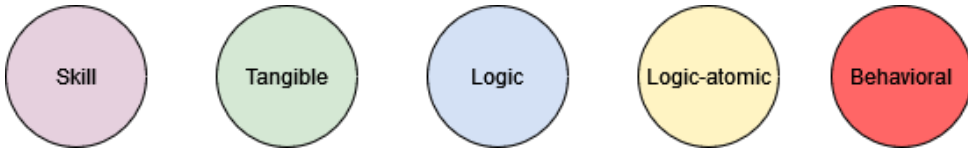


Figure 3.3: Resource alternatives, and associated colour

5. Behavioral: Behavioral resources are actions an attacker must complete to make an attack. Behavioral actions include a user opening a phishing email or an attacker plugging in a USB drive.

3.5.4 Estimating Cost

The RCM model states that to launch an attack, the attacker must have completed all seven stages of the kill chain. Hence the total cost of an attack is the sum of all the required resources from each stage [Hag20]. Each resource alternative is therefore assigned a minimum cost, a maximum cost, and a confidence parameter. The user assigns the confidence parameter a value between 0 to 1. A confidence of 0 states that the model has almost no evidence to support the stated cost interval and several uncertain factors are involved. If the confidence is set to 1, however, the user has evidence to back his claim and that the cost of the resource alternative is not subject to significant variation.

V

Represents the set of resources alternatives associated with one step of the kill chain

α

Represents the minimum estimated cost for the cheapest resource alternative associated with a resource

β

Represents the maximum cost of the most expensive resource alternative associated with a resource

θ_j

Average confidence of the resource alternatives associated with resource j .

$$Estimated_Cost = [MinCost, MaxCost, Confidence] \quad (3.1)$$

$$minimum_cost = \sum_{stage \in kill\ chain} \sum_{i \in V} \alpha_i \quad (3.2)$$

$$maximum_cost = \sum_{stage \in kill\ chain} \sum_{i \in V} \beta_i \quad (3.3)$$

$$\theta_j = (\sum_{i \in R_j} c_j) / n \quad (3.4)$$

$$confidence = \prod_{stage \in killchain} \prod_{j \in R} \theta_j \quad (3.5)$$

3.5.5 STRIDE

The STRIDE methodology identifies potential cyber security systems threats and serves as a mnemonic for the six categories of threats as seen in Table 3.1 [KMLS17]. The attributes seen in STRIDE can be used to reason about the system’s threats. The strength of STRIDE is that it is general in nature, so it is possible to tailor it to specific situations. Looking at each attribute lets us reason about how spoofing, tampering, repudiation etc., impacts each system. Using these as a starting point for our analysis, we can backtrack and attempt to identify what tactics and methods a malicious agent could employ to pose these threats to the system.

Table 3.1: Overview of security properties violated by STRIDE (adapted from [Poo20])

STRIDE attribute	Violated property	Description
Spoofing	Authenticity	The act of falsifying data, for instance, to impersonate a victim.

Continued on next page

Table 3.1: Overview of security properties violated by STRIDE (adapted from [Poo20]) (Continued)

STRIDE attribute	Violated property	Description
Tampering	Integrity	Attacker manipulates data, for instance, by intercepting a message and altering its contents before forwarding it to the recipient.
Repudiation	Non-reputability	Ability to deny that an action was taken, for instance, that a message was sent.
Information Disclosure	Confidentiality	Violation of security where a malicious party achieves access to confidential data.
Denial of service	Availability	The act of making a service unavailable to other legitimate users by overloading the system and reducing the remaining resources.
Elevation of privilege	Authorization	Gaining privileges or elevated access to resources which is normally protected, for instance, by exploiting bugs or design flaws in the system.

Chapter 4

Results from interviews

In order to answer our research questions, we have looked at the security of VDES and its predecessor AIS from multiple perspectives. The literature review enabled us to see these systems from the perspective of researchers in the field of maritime security. Hence, we have taken the perspective of a malicious actor in our experiments and analysis, to get a more nuanced view. To draw a complete picture of the overall security, we also have to investigate our research questions from the perspective of the developers and service providers that ultimately deliver the service to end users. This chapter presents findings from our three interviews involved in developing VDES.

4.1 Interview questions

During the interviews, we chose to talk about the following three topics in accordance with our research questions:

1. What is the general perception of the cyber security level in the maritime industry? Are you aware of the existing vulnerabilities in AIS and GNSS?
2. How do you expect VDES will be utilized in the maritime shipping industry? Which services do you think will be used over VDES?
3. Do you think the introduction of VDES will improve the cyber security within the maritime shipping industry? If yes, how and why?

We chose these questions specifically to guide the discussion with our interview subject in the direction of their perception of security and how they envision this to evolve with the introduction of VDES.

4.1.1 Cyber Security practices, and known AIS attacks

Among the interviewees, everyone was aware of the vulnerabilities in AIS, and cases of misuse, such as vessels turning off their equipment, were mentioned. Since AIS

messages are easily forged, they are hard to verify. Because of this, ships use various tools, such as radar, and Maritime administrations do not perceive AIS as secure. This ensures that AIS is not used as a stand-alone tool but rather used to complement information obtained by radar or other equipment. An example that reoccurred was that even though AIS was turned off, radar would detect the ship. End users of the system do, however, think of AIS as useful to gain information about nearby vessels. Therefore, how the receiver uses AIS is crucial. Across all our interviews, the consensus is that AIS is not to be trusted and that received data should not be used in a closed-loop system.

Additionally, authentication as a mitigation technique was also mentioned by all our participants as a necessary mechanism to improve the current situation. Furthermore, VDES was mentioned by one as a prerequisite for enabling authentication in AIS. However, they were also aware of the implementational challenges of a maritime PKI and stated that this would likely not be mandatory for all AIS messages.

Another challenge identified was the standardization lifecycle in international bodies such as IMO, where cycles of up to 4 years could be expected to update standards. Challenges in the implementation of VDES are also present, and one of our interviews pointed out how different components could be poorly implemented. Most importantly, there is a possibility that hidden issues are not covered by testing, further substantiated by the increase in complexity.

4.1.2 How will VDES be utilized, and which services will it be used for

A consensus identified is that AIS is a well-recognized and important tool for safety at sea. It is, therefore, important that VDES continues to be compatible with the legacy system, such as AIS. The introduction of VDE channels, such as VDE-TER and VDE-SAT, enables new services to be provided while ships are out of port. Overall, VDES enables many services which have previously not been possible.

During one of our interviews, VDES R-mode was mentioned as a positive addition, which enables vessels to derive their position by being within the coverage of VDES base stations when GNSS alternatives are unavailable, for instance during a GNSS spoofing attack. Another interview identified Maritime Safety Information (MSI) services as a critical driver for VDES development, which would aid VTSs to manage ports and have a positive impact on numerous areas such as route optimization and route exchange for SART operations. One of the main benefits is reduced CO_2 emissions since routes become more efficient and traffic control becomes more streamlined. When ships can plan their arrival according to other ships, they can manage their speed more sustainably. This would improve the current situations

where ships travel at high speed and arrive early at their destination but have to wait in line. Even saving fuel consumption by 1% can significantly impact the industry’s environmental effects if, instead, ships arrive just in time. Our final interview also identified benefits related to mandatory ship reporting towards VTSs and tug boat services, which today is carried over voice channels. In high-traffic areas such as Singapore, vessels could use many hours traversing up to 16 VTS zones, which is another application for VDES based services.

Regarding what future services might be serviced over VDES, one of the interviewees mentioned the S-100 standard and its associated products. Two of the interviewees mentioned that only approved organizations would have the possibility of registering as a service provider at an MCP instance. This reduces the probability of malicious service providers and the risk of receiving malicious or fraudulent data.

4.1.3 VDES security improvements

On the topic of security improvements in VDES, the general perception was that there is increased awareness of cyber security within the industry. However, the benefits must outweigh the costs, making it challenging to standardize and enforce as mandatory. Consequently, the development of VDES needs to be pushed by consumer demand, who are also looking at saving money where possible. However, they might not be able to justify the extra cost associated with authentication mechanisms. Currently, the certificate revocation problem is one of the major challenges of implementing a PKI. However, some pointed out that this is a “luxury” problem, as the risk of someone exploiting this vulnerability increases with the adoption rate of VDES. In other words, if it is worth attacking, it is also worth protecting. For example, installing more base stations reduces vessels’ offline time. However, this is not cost-effective before the number of users has reached a certain point.

One of the interviewees also mentioned that authentication is not always easy, regardless of the PKI infrastructure. For the receiver to verify the signature of the data, the number of bit errors needs to be minor, and Forward Error Correction (FEC) is necessary. During this conversation, the TESLA protocol was also mentioned to be employed as a way to authenticate AIS messages by sending the initial AIS auth message over VDE and let the subsequent authenticated traffic go over AIS channels.

4.2 Key takeaways

From our interviews, we found that our interviewees had similar perceptions on both security and application capabilities of VDES. One of the key takeaways from our interviews is that VDES brings several benefits to the table and is a much-needed

upgrade compared to the current situation. Each interviewee was aware of challenges regarding authentication and implementing a PKI, which is currently one of the leading security concerns. Because VDES is not yet mandatory, the adoption rate of VDES was identified as a critical driver in development. Finally, ship manufacturers and shipping companies must be convinced of the benefits of VDES and that the value of future maritime services outweighs the costs.

Chapter 5

Results from AIS Experiment

5.1 Introduction

This chapter goes into detail on our AIS-based attacks, and the steps we took to create our AIS frame builder in Python. Our experiment used two SDRs, one functioning as a transmitter and the other as a receiver. SDRs were chosen for this purpose primarily due to the high amount of open-source code available.

The widespread knowledge that SDR can effectively transmit, and receive AIS messages played a significant role in our decision to employ them in our experiment [WH20; BPW14; For22]. By offering an explanation of the tools and techniques required to carry out AIS-based attacks, this chapter aims to provide valuable insights into how an adversary can exploit vulnerabilities in AIS and use Python-embedded blocks in GNU Radio to customize attacks.

5.2 Motivation and assumptions

The motivation behind conducting the AIS experiments is to investigate how the attack surface of AIS systems have evolved since the work carried out by [WH20]. The experiment assumes that TDMA slot allocation is handled separately. This is a fair assumption, as this does not affect attacks related to positioning. Therefore, we made the same assumption for our experiments and focus on forging and validating AIS messages.

The main motivation behind reconstructing the experiment is to verify that it is still possible and to investigate how the latest advancements in software and hardware impacts the feasibility of the attacks. These results was valuable in attempting to answer our research questions.

5.3 Equipment

This section lists the software and hardware tools used in our experiment.

- Computer
- Windows 11
 - OS type: 64-bit
 - Memory: 500 GB
- The experiment ran on two identical Virtual Machines
 - Ubuntu 20.04.5 LTS
 - OS type: 64-bit
 - Processors: 2
 - Base Memory 2412 MB
- Transmitter SDR
 - HackRF One
 - * half-duplex transceiver
 - * USB-powered
 - * 1 MHz to 6 GHz operating frequency
 - * compatible with GNU radio
- Receiver SDR
 - USRP b200mini
 - * full duplex
 - * USB powered
 - * 70 MHz to 6 GHz operating frequency
 - * compatible with GNU radio

- Wired Communication
 - wire
 - 10 dB Attenuator

- Software
 - Transmitter
 - * GNU radio
 - * AIS_TX
 - * gr-osmosdr
 - * hackrf

 - Receiver
 - * GNU radio
 - * gr-ais
 - * Socat
 - * Opencpn
 - * UHD

5.3.1 USRP B200mini

The USRP B200mini is a SDR with a wide frequency range, ranging from 70 MHz to 6 GHz [Bra]. This makes it suitable to receive and transmit AIS messages, which are sent at 161.975 and 162.025 MHz [ITU22]. It is also Universal Serial Bus (USB)-powered, using USB 3.0, and is compatible with GNU radio, which made it a natural choice for the AIS experiment [Bra]. The USRP B200 was also used by Walde and Hanus at the receiver end [WH20], making it a suitable SDR for our experiment.

5.3.2 HackRF One

The HackRF One is a SDR developed by Great Scott Gadgets. It can transmit or receive radio signals from 1 MHz to 6 GHz, and it is USB powered and compatible with GNU radio [Gre]. We decided to use the HackRF One as a transmitter, as Walde and Hanus proved that the HackRF One could be used as a transmitter [WH20].



Figure 5.1: USRP and HackRF One used in the experiment

5.3.3 GNU radio

GNU Radio is a free and open-source toolkit that provides signal processing blocks to implement SDRs [GNU22]. Examples of processing blocks are modulators, de-

modulators, and sinks, to name a few. The blocks can then be connected using a user-friendly drag-and-drop interface. Another essential feature of GNU Radio is that it is possible to create and add blocks that are not already defined. This can, for example, be done by utilizing the Python embedded block, which can be customized to the users' needs.

5.3.4 AIS_TX

AIS_TX is an AIS transmitter designed and implemented as SDR by Balduzzi [BPW14]. AIS_TX is built on top of GNU Radio and contains a block called AIS Frame Builder, which takes Automatic Identification System Vessel Data Message (AIVDM) encoded sentences as input, builds an AIS frame, and transmits it on AIS channel A or B. AIS_TX can be run through the Command Line Interface (CLI) or in GNU Radio Companion. The Original AIS_TX setup is shown in Figure 5.2.

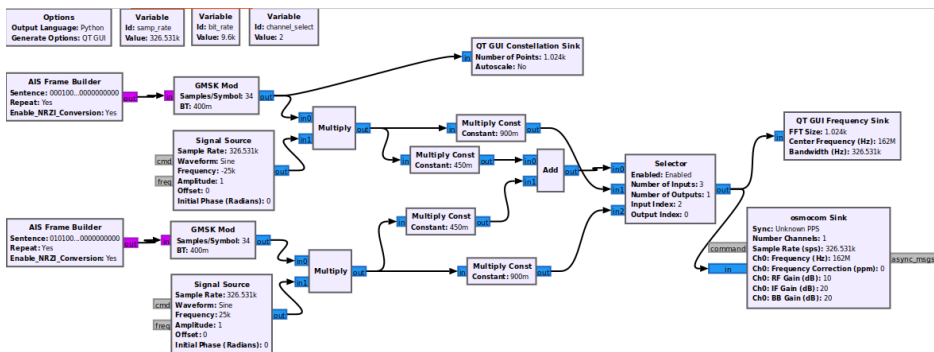


Figure 5.2: Original AIS_TX setup by [BPW14]

5.3.5 gr-ais

GNU Radio Automatic Identification System (gr-ais) is an open-source software program designed to receive AIS signals from ships and shore stations [Bis21]. gr-ais is built on top of GNU Radio and processes radio signals from nearby AIS transponders [Bis21]. Used with OpenCPN, this feature helps users track and display ship positions, movements, and other information transmitted with AIS. In this experiment, gr-ais was used on the receiver site to detect and decode transmitted messages.

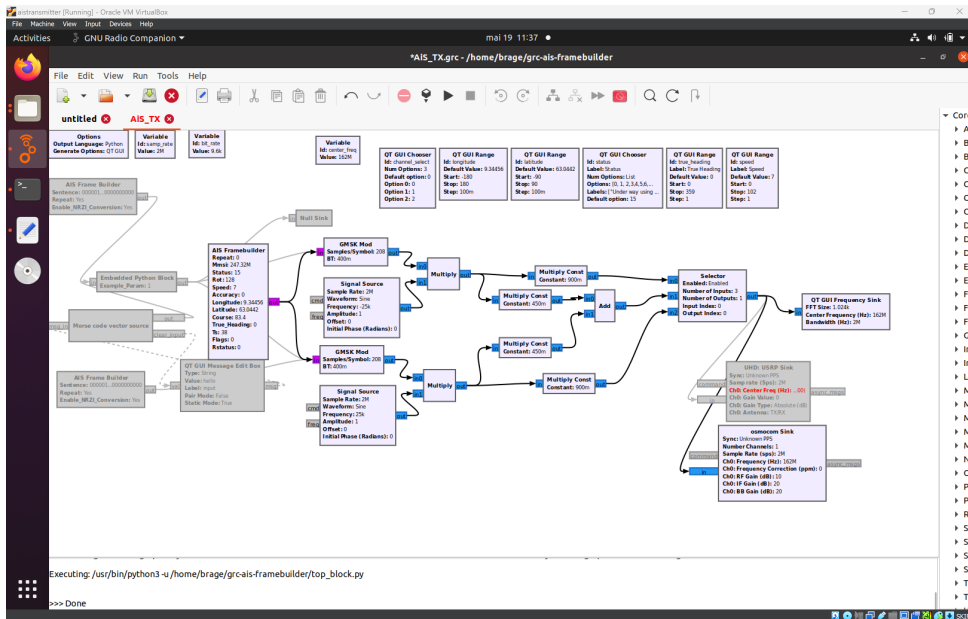


Figure 5.3: AIS_TX setup with our own modifications [BPW14]

5.3.6 gr-osmosdr

gr-osmosdr is an open-source software package developed on the GNU Radio Framework. The package includes blocks designed to interface with various SDRs such as the HackRF One [Osm23]. Since GNU Radio does not inherently support radio hardware like the HackRF One, the gr-osmosdr package is essential for utilizing the HackRF as a transmitter.

5.3.7 OpenCPN

Open Chart Plotter Navigator (OpenCPN) is an open-source marine navigation software. OpenCPN offers chart plotting, route planning, and GPS tracking features [Ope]. In this experiment, OpenCPN was used to visualize AIS messages by integrating AIS data into its chart plotting interface.

5.3.8 Socat

Using Socat, it's possible to establish a relationship between two data sources to create bidirectional data streams [Deb; WH20]. This experiment used Socat to link gr-ais and OpenCPN. By utilizing Socat to establish a data stream between gr-ais and OpenCPN, we were able to visualize the AIS data decoded by gr-ais in OpenCPN.

5.4 Set up Transmitter

The following steps were conducted to set up the transmitter. The operations described follow a combination of the installation guidelines described by Walde and Hanus, and Forsberg [WH20; For22]. On the transmitter side, we decided to use the HackRF as the transmitter. To set up the transmitter, we had to install the HackRF software. This was done with the command in Listing 5.1.

Listing 5.1: Installing hackrf, adapted from [For22]

```
#!/bin/bash
sudo apt-get install hackrf
sudo apt-get install libhackrf-dev
```

For GNU Radio, we decided to use version 3.8. The installation steps are shown in Listing 5.2

Listing 5.2: Installing GNU Radio, adapted from [For22]

```
#!/bin/bash
sudo add-apt-repository ppa:gnuradio/gnuradio-releases-3.8
sudo apt-get update
sudo apt-get install gnuradio python3-packaging
```

Since we decided to use the HackRF as the transmitter, we had to switch the "UHD: USRP Sink" block with an osmocomblock. The following steps were done to install gr-osmosdr for GNU Radio release 3.8.

Listing 5.3: Installing gr-osmosdr, adapted from [WH20]

```
#!/bin/bash
git clone https://github.com/osmocomblock/gr-osmosdr.git
cd gr-osmosdr
git checkout cffef69
mkdir build
cd build
cmake ..
make
sudo make install
sudo ldconfig
```

Forsberg also discovered that the command in Listing 5.4 had to be added in `./profile` for GNU radio to use the custom blocks [For22].

Listing 5.4: New python path environment variable, adapted from [For22]

```
export PYTHONPATH=/usr/local/lib/python3/dist-packages:$PYTHONPATH
```

For our experiment, we ended up using the Python version of the AIS frame builder, and the installation guide is shown in ¹. The code used in the Python block can be found in the mentioned GitHub repository² and is called *AIS_Framebuilder.py*. Copying and pasting the code from GitHub into a GRC Python block is sufficient to import the frame builder. We have also included the GRC flow graph itself, called *AiS_TX.grc*, which can simply be opened in GRC.

5.5 Set up Receiver

The USRP B200mini was chosen as our receiver, and the necessary software had to be downloaded to connect the USRP via USB. UHD is part of the package management on Ubuntu and can be installed as shown in Listing 5.5³.

Listing 5.5: Installing UHD

```
#!/bin/bash
sudo apt-get install libuhd-dev uhd-host
```

Since we intended to use gr-ais, we needed a GNU Radio release compatible with gr-ais. Because we already were using release 3.8 on the transmitter side, it made sense to do the same on the receiver side. The installation is shown in Listing 5.2. The commands for installing gr-ais are shown in Listing 5.6.

Listing 5.6: Installing gr-ais, adapted from [For22]

```
#!/bin/bash
sudo git clone https://github.com/bistromath/gr-ais.git
sudo apt-get install liborc-0.4-dev
cd gr-ais
mkdir build
cd build
cmake -DCMAKE_INSTALL_PREFIX=/usr ..
make
sudo make install
sudo ldconfig
```

As on the transmitter side, the command in Listing 5.4 was added in `./profile` for GNU radio to use the custom blocks [For22]. Socat and OpenCPN was then installed using the commands shown in Listing 5.7 and Listing 5.8

¹https://wiki.gnuradio.org/index.php/Embedded_Python_Block

²<https://github.com/collinlokken/grc-ais-vdes-framebuilder>

³https://files.ettus.com/manual/page_install.html

Listing 5.7: Installing Socat, adapted from [WH20]

```
#!/bin/bash
sudo apt-get install socat
```

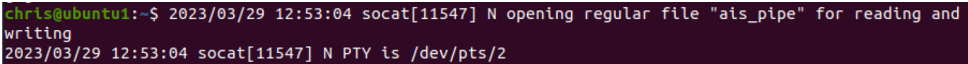
Listing 5.8: Installing OpenCPN, adapted from [WH20]

```
#!/bin/bash
sudo add-apt-repository ppa:opencpn/opencpn
sudo apt-get update
sudo apt-get install opencpn
```

The next step is to start listening to AIS transmissions. In order to connect gr-ais with OpenCPN, we started with setting up Socat as shown in Listing 5.9. The output is shown Figure 5.4.

Listing 5.9: Set up a virtual serial port to read from, adapted from [WH20]

```
#!/bin/bash
socat -d -d pipe:ais_pipe pty&
```



```
chr1s@ubuntu1:~$ 2023/03/29 12:53:04 socat[11547] N opening regular file "ais_pipe" for reading and
writing
2023/03/29 12:53:04 socat[11547] N PTY is /dev/pts/2
```

Figure 5.4: Socat Output

As evident from the output shown in Figure 5.4, the virtual serial port in this instance is `/dev/pts/2`. To establish a connection between OpenCPN and gr-ais, it is necessary to add this virtual serial port to OpenCPN. This can be done by adding it in the graphical interface of OpenCPN as shown in Figure 5.5. In this case `/dev/pts/2`, the virtual serial port is displayed under the Serial Port section.

With the necessary configurations complete, it's time to launch gr-ais. gr-ais, is launched with the command shown in Listing 5.10

Listing 5.10: Start gr-ais, adapted from [WH20]

```
#!/bin/bash
ais_rx -suhd -g35 -r250e3 > ais_pipe
```

5.6 AIS_TX Implementation

This section discusses how the AIS_TX module was implemented in Python, including some of the challenges we experienced. Using the original frame builder as inspiration was a good starting point for porting the software to Python.

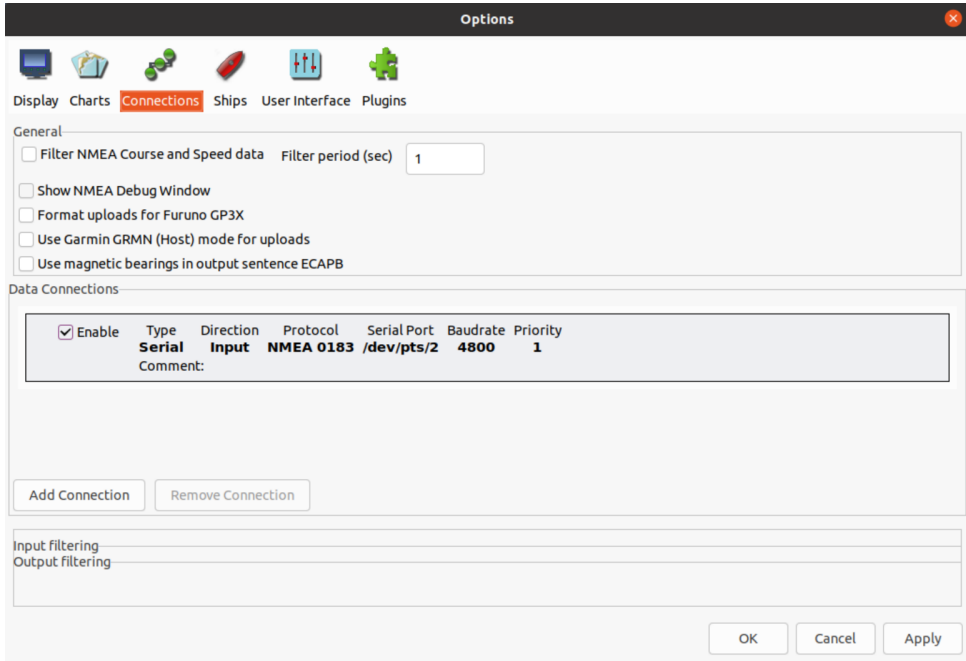


Figure 5.5: Set up OpenCPN connection

Since the introduction of GRC 3.8, support for Python blocks has become available, enabling the development of custom blocks. We, therefore, decided to create our own AIS frame builder in Python, building on top of the works carried out by Balduzzi et al. [BPW14]. The main motivation behind this was that we wanted to set the values at runtime, which was impossible with the current implementation. The custom python blocks interface well with the other Qt Graphical User Interface (GUI) widgets in GRC, which enables us to change values such as ship position quickly. The code for our custom block as well as the setup in GRC, can be found in our GitHub repository [LB23].

The original AIS_TX module written in C++ by [BPW14] included an AIVDM encoder tool that would convert the desired vessel values to a bit-represented payload. This payload then had to be copied over to the AIS_TX module, either via the GRC GUI or via CLI. Our first approach was to substitute the AIVDM encoder tool by instead using GRC Qt GUI elements which we could control at runtime. This approach, unfortunately, did not pay off, as the AIS_TX block only read the GUI element values once, and we had no control over the input beyond that point. We found this easier in Python, where no compilation or installation is required as it is an interpreted language.

The original flowgraph is shown in Figure 5.2, and the flowgraph with our modifications is shown in Figure 5.3. We focused on assigning the AIS message values shown in Table 5.1 to show what is possible to achieve with this. These are of specific interest to an attacker, as it becomes easier to simulate the movement of the ship.

Table 5.1: Values that we chose to focus on for our experiment

AIS values	Description
True Heading	The direction the ship is pointing
Course	Where the ship is trying to reach
Speed	The speed of the ship in knots
Longitude, Latitude	Coordinates of the ship

5.6.1 Frame creation

In this section we explain the technical details of the frame builder, and how the input values are processed into a frame. The creation of frames goes through the following procedure:

1. Byte representation
2. CRC calculation
3. Endianness
4. Bit stuffing
5. Padding
6. Nrz to NRZI

Byte representation

Since the inputs to our frame builder are integers, these have to be concatenated and turned into bytes to continue the processing. To achieve the desired output, the integers must first be turned into a string of bits and then padded to the left with zeros. For instance, the vessel speed in knots could be 10 knots, which is turned into the string “1010” and padded to fill the remaining bits (“0000001010”), with a total of 10 bits. The final 168-bit payload is then turned into bytes before being passed to the next stage.

0x0000	0x1189	0x2312	0x329B	0x4624	0x57AD	0x6536	0x74BF
0x8C4B	0x9DC1	0xAF5A	0xBED3	0xCA6C	0xDBE5	0xE97E	0xF8F7
0x1081	0x0108	0x3393	0x221A	0x56A5	0x472C	0x75B7	0x643E
0x9CC9	0x8D40	0xBFDB	0xAE52	0xDAED	0xCB64	0xF9FF	0xE876
0x2102	0x308B	0x0210	0x1399	0x6726	0x76AF	0x4434	0x55BD
0xAD4A	0xBCC3	0x8E58	0x9FD1	0xEB6E	0xFAE7	0xC87C	0xD9F5
0x3183	0x200A	0x1291	0x0318	0x77A7	0x662E	0x54B5	0x453C
0xBDCB	0xAC42	0x9ED9	0x8F50	0xFBEF	0xEA66	0xD8FD	0xC974
0x4204	0x538D	0x6116	0x709F	0x0420	0x15A9	0x2732	0x36BB
0xCE4C	0xDFC5	0xED5E	0xFCF7	0x8868	0x99E1	0xAB7A	0xBAF3
0x5285	0x430C	0x7197	0x601E	0x14A1	0x0528	0x37B3	0x263A
0xDCCD	0xCF44	0xFDDF	0xEC56	0x98E9	0x8960	0xBBFB	0xAA72
0x6306	0x728F	0x4014	0x519D	0x2522	0x34AB	0x0630	0x17B9
0xEF4E	0xFEC7	0xCC5C	0xDDD5	0xA96A	0xB8E3	0x8A78	0x9BF1
0x7387	0x620E	0x5095	0x411C	0x35A3	0x242A	0x16B1	0x0738
0xFFCF	0xEE46	0xDCDD	0xCD54	0xB9EB	0xA862	0x9AF9	0x8B70
0x8408	0x9581	0xA71A	0xB693	0xC22C	0xD3A5	0xE13E	0xF0B7
0x0840	0x19C9	0x2B52	0x3ADB	0x4E64	0x5FED	0x6D76	0x7CFF
0x9489	0x8500	0xB79B	0xA612	0xD2AD	0xC324	0xF1BF	0xE036
0x18C1	0x0948	0x3BD3	0x2A5A	0x5EE5	0x4F6C	0x7DF7	0x6C7E
0xA50A	0xB483	0x8618	0x9791	0xE32E	0xF2A7	0xC03C	0xD1B5
0x2942	0x38CB	0x0A50	0x1BD9	0x6F66	0x7EEF	0x4C74	0x5DFD
0xB58B	0xA402	0x9699	0x8710	0xF3AF	0xE226	0xD0BD	0xC134
0x39C3	0x284A	0x1AD1	0x0B58	0x7FE7	0x6E6E	0x5CF5	0x4D7C
0xC60C	0xD785	0xE51E	0xF497	0x8028	0x91A1	0xA33A	0xB2B3
0x4A44	0x5BCD	0x6956	0x78DF	0x0C60	0x1DE9	0x2F72	0x3EFB
0xD68D	0xC704	0xF59F	0xE416	0x90A9	0x8120	0xB3BB	0xA232
0x5AC5	0x4B4C	0x79D7	0x685E	0x1CE1	0x0D68	0x3FF3	0x2E7A
0xE70E	0xF687	0xC41C	0xD595	0xA12A	0xB0A3	0x8238	0x93B1
0x6B46	0x7ACF	0x4854	0x59DD	0x2D62	0x3CEB	0x0E70	0x1FF9
0xF78F	0xE606	0xD49D	0xC514	0xB1AB	0xA022	0x92B9	0x8330
0x7BC7	0x6A4E	0x58D5	0x495C	0x3DE3	0x2C6A	0x1EF1	0x0F78

Figure 5.6: CRC lookup table for 16-bit polynomial

CRC calculation

The CRC calculation uses a 16-bit polynomial in accordance with ISO/IEC 13239:2002 and uses an initial value of $0xFFFF$ (16 1-bits). The calculation follows a simple procedure that involves looking up the corresponding CRC division values from the lookup table, shown in Figure 5.6. This table is a precomputed set of the remainder after the division done by the CRC algorithm for each possible byte, which helps optimize the CRC computation [Sun23].

Endianness

After the CRC-16 check sequence has been appended, the order of each byte should be output with the least significant bit first, as shown in Figure 5.7 [ITU14]. The preamble, start flag and flag are also appended with the same endianness.

Bit stuffing

Bit stuffing is done to break up large groups of 1-bits in a data frame by inserting 0s into the data stream. An algorithm that counts consecutive ones and inserts 0-bits

Original Bit Order	AAAAABC	DDDDDDDD	DDDDDDDD	DDDDDDDD	DDDEE000
After Endianness	CBAAAAA	DDDDDDDD	DDDDDDDD	DDDDDDDD	000EEDDD

Figure 5.7: Output order of each byte, adapted from [ITU14]

when this is detected must be employed to achieve this. This is done to both the data payload and CRC check sequence.

Padding

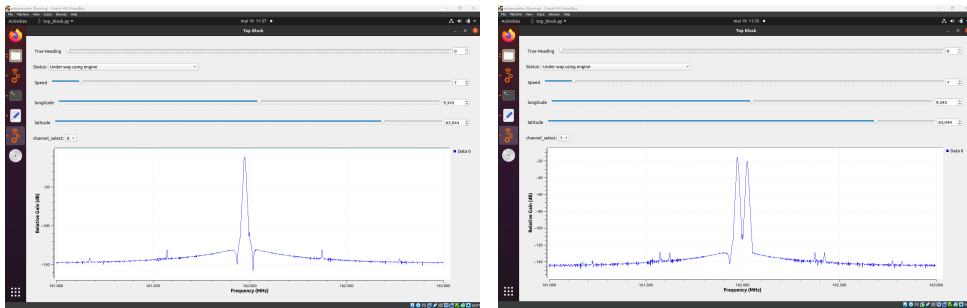
To fill up the 256 bits of the total frame length, zeros are appended to the end of the frame. The padding length can vary based on factors such as how many bits were inserted during the bit-stuffing phase.

NRZ to NRZI

Finally, the [ITU14] also states that NRZI encoding is needed, making transmission easier for the underlying hardware. In this version of NRZI encoding, the zero bits act as a switch that changes the voltage. When a 0-bit is encountered in the input, the following output bit should be the opposite of the last. The next output bit is not changed when 1-bits are encountered in the input. For example, 01001110 becomes 11011110.

5.6.2 AIS_TX as GRC block

In GRC, the data flow is handled via input and output buffers, in which data can be read and processed data can be written too. Since the frame builder would be the first block in the graph, no input buffers were needed. In our code, this is expressed as *None*. After creating the frame, it should be forwarded to the next GRC block in the flow graph via the output buffer. The documentation on how this should be done is lacking, and some time was used in debugging. The output buffer is a NumPy array of size 32768, filled with only zeros, which we overwrite with our data of 32 bytes. Because we also set the output type to bytes, GRC expects one byte at each index. For the output buffer to receive data, the length of the desired output must be set correctly. If not, the entire buffer is interpreted as valid data, even though our data only made up 0.01% of the buffer. The length returned should be the portion of the buffer that contains valid data, so the first 32 in our case.



(a) One channel enabled

(b) Both channels enabled

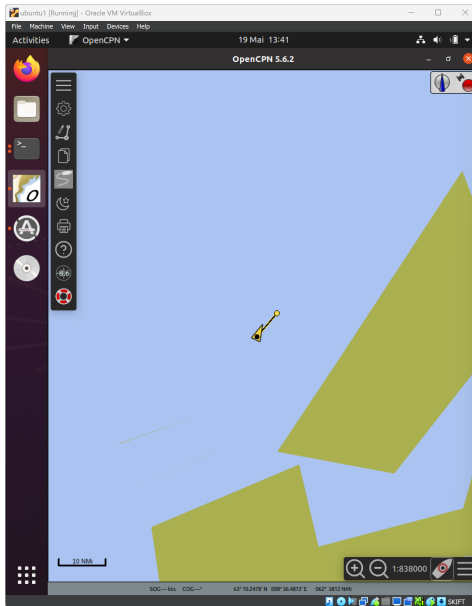
Figure 5.9: AIS TX control panel

5.7.1 Main benefits

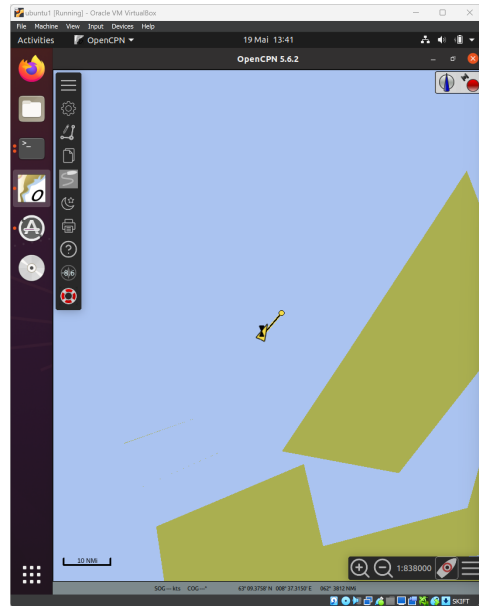
After observing the results in OpenCPN, we can confidently say that we managed to implement a more flexible AIS frame builder. Since it is now possible to implement blocks directly into GRC, we found a much easier way of assigning message values. We also found Python to be the correct choice of programming language. It is also very popular and is considered to be beginner-friendly. Providing the frame builder in this format decreases the entry cost for someone that wishes to continue evolving and customizing the tool. Another benefit of using Python is that customization of the code and checking if it yields the desired output is quicker since Python is interpreted.

In our implementation, we focused on a handful of values. However, all the other AIS message values can also be assigned for all message types. It is also possible to create scripts that, for instance, take a desired arrival location as input and outputs simulated movement patterns towards it. This makes it possible to incorporate physics and the physical attributes of the ship, such as weight, draught, rate of turn, etc. This opens up various possibilities, as we can simulate more complex attacks that incorporate vessels' movement over time. The main benefits that we observed during our experiment are:

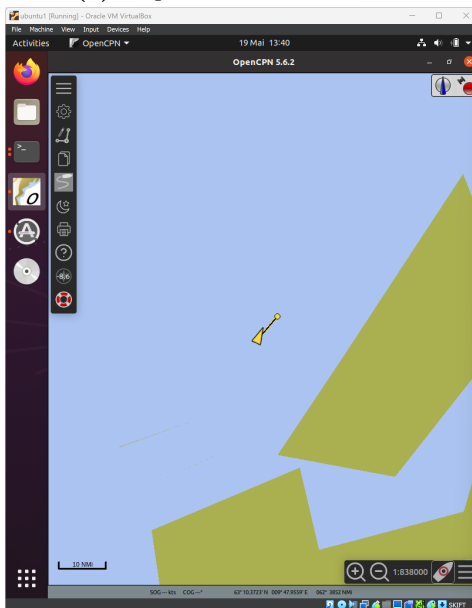
1. Greater room for customization
2. Enable more elaborate attacks
3. Lower implementation cost, as installation of the custom block is no longer needed
4. Easier to debug



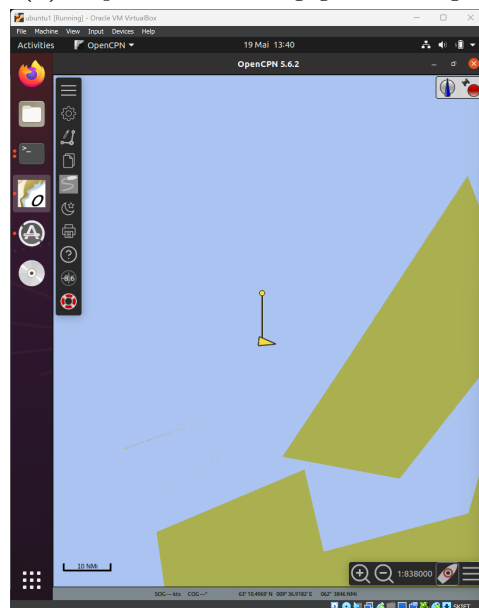
(a) Ship with status "Moored"



(b) Ship with status "Engaged in fishing"



(c) Ship with lower speed, course, and heading northwest



(d) Ship with higher speed and course straight north, heading west

Figure 5.10: OpenCPN output

Chapter 6

Results from VDE-TER Experiment

If an adversary wants to transfer messages over VDE-TER, a VDE-TER signal simulator, similar to AIS_TX, is needed. Our AIS experiment proved that the AIS frame builder block could be implemented as a Python block in GNU Radio, which enables greater room for customization. This chapter contains the necessary steps to build what we imagine a possible VDE-TER frame builder and signal simulator might look like. We hope the knowledge presented in this chapter can help lay the foundation and fuel further exploration and innovations in the open-source world of VDES.

6.1 VDE-TER Frame builder

To transmit messages over VDE-TER, the first step is to make a frame builder according to the technical specification in the ITU-R M.2092-1 [ITU22]. From the information obtained in the technical specification for VDES, we developed a flow diagram with the necessary steps to build a VDE-TER compatible frame. The flow diagram is shown in Figure 6.1. The following section only consists of a fraction of the VDES specification, and it is recommended to read the technical specification of VDES ¹ to get a more detailed view.

¹https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2092-1-202202-1!!PDF-E.pdf

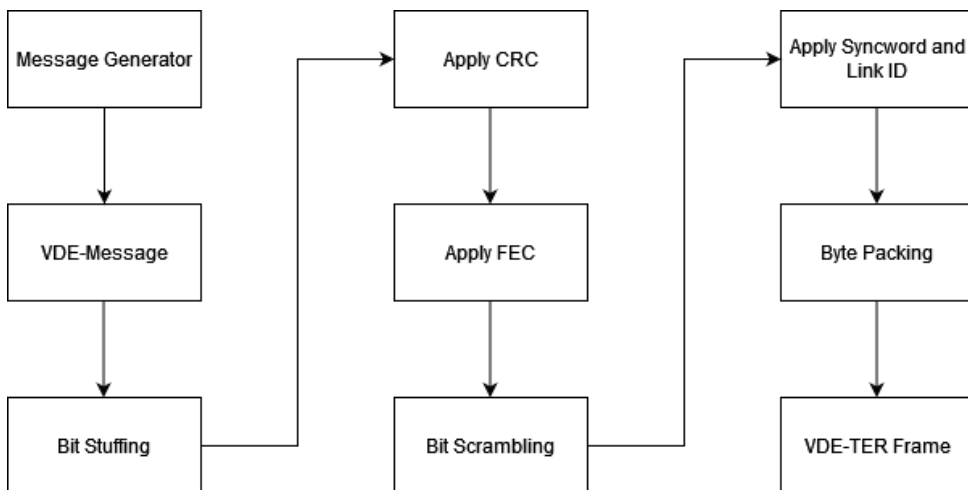


Figure 6.1: A VDE frame builder (adapted from [ITU22])

6.1.1 Generate a VDE-TER message

In the technical specification of VDES, different messages designed for VDE-TER are defined in Annex 4 [ITU22]. One of these messages is the short data message, which should be transmitted on the Random Access Channel (RAC) when used in ship-to-ship communication. Because short data messages can be sent without a resource allocation, we decided to focus on it to simplify the process as much as possible.

Table 6.1: Short data message with acknowledgment, adapted from [ITU22])

Function	Size in Bytes	Description
Type	1	92
Length	2	Total size in bytes
Source ID	4	Unique 9 digit unique identifier as explained in ITU-R M.585-9
Session ID	1	Set to 0, reserved for future use
Destination ID	4	MMSI of the receiving ship
Retransmission no	1	Value between 0 and 255, increment with every transmission
Payload	Variable	The payload to be transmitted

6.1.2 Bit Stuffing

As VDE-TER packets always shall fit into one slot, and the number of bits transmitted per VDE-TER packet shall be fixed depending on the Link ID, it is necessary to append zero padding to the end of the VDE-TER message. It is not stated exactly how long each VDE-TER packet is, but by subtracting 32 from the total FEC input bits, we can compute the length of the VDE-TER packet and append zeros accordingly. Since the FEC input bits vary between different Link IDs, this has to be calculated for each Link ID respectively.

6.1.3 Compute and apply CRC32

In VDE-TER, a CRC 32-bit check sequence is appended to the last segment of the datagram. The CRC-32 is calculated with the generator polynomial shown in Equation (6.1).

$$x^{32} + x^{26} + x^{23} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1 \quad (6.1)$$

The CRC check sequence is the remainder of the datagram divided by the generator polynomial. According to ITU, this can effectively be calculated by applying a linear feedback shift register with the initial value $0xFFFFFFFF$ [ITU22]. For well-known polynomials, hard-coded lookup tables exist. In this case, the polynomial is $0x04C11DB7$. We computed the CRC-32 using well-known formulas and a known lookup table adapted from ^{2,3}. After the CRC-32 check sequence value is calculated, it is appended at the end of the datagram.

6.1.4 Forward Error Correction

After the CRC-32 check sequence has been appended, FEC is applied using turbo encoding. The overall structure follows the specification in (ETSI) EN 302 583⁴ [ITU22]. The overall structure of the FEC encoder is shown in Figure 6.2.

²<https://github.com/Michaelangel007/crc32>

³<https://gist.github.com/Miliox/b86b60b9755faf3bd7cf>

⁴https://www.etsi.org/deliver/etsi_en/302500_302599/302583/01.02.01_40/en_302583v010201o.pdf

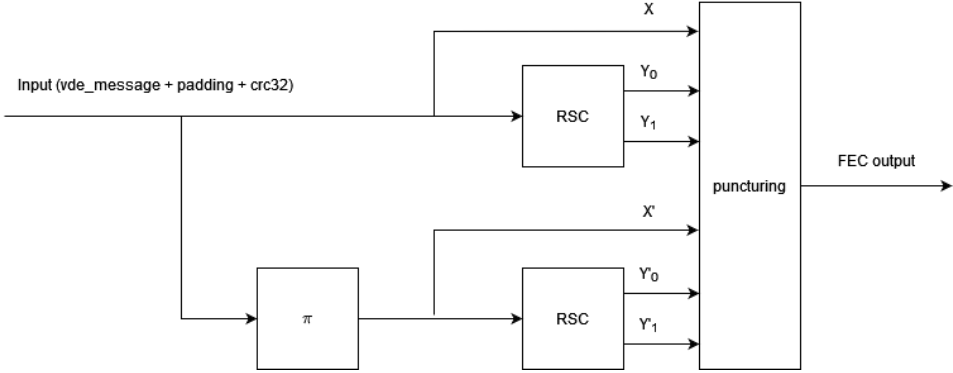


Figure 6.2: General encoder structure, (adapted from [ITU22])

Interleaver

In Figure 6.2, the interleaver block is shown as the π block. In the interleaver block, the input data is reordered. The permutation numbers are calculated using the operations shown in Equation (6.2) [ITU22].

$$\begin{aligned}
 m &= (s - 1) \bmod 2 \\
 i &= \text{floor}((s - 1)/(2 * k_2)) \\
 j &= \text{floor}((s - 1)/2) - i * k_2 \\
 t &= (19 * i + 1) \bmod (k_1/2) \\
 q &= t \bmod 8 + 1 \\
 c &= (p_q * j + 21 * m) \bmod k_2 \\
 \pi(s) &= 2(t + c * k_1/2 + 1) - m
 \end{aligned} \tag{6.2}$$

In Equation (6.2), $s \in (1, \dots, k)$, where $k = k_1 * k_2$. p_q , $q \in (1, \dots, 8)$, and p is a list of primes associated with a Link ID. Different Link IDs are associated with different values. In Table 6.2, parameters for a couple of Link IDs are provided. The permutation numbers shall be interpreted such that the s^{th} bit read out after interleaving is $\pi(s^{th})$ bit of the information block [ITU22]. As an example, if $s = 1$, and $p_i(1) = 2$, then $\text{input_data}(2)$ should be the first bit out of the interleaver block.

Table 6.2: Link IDs and their associated parameters for interleaving, (adapted from [ITU22])

LinkID	k1 k2	p
5	2 144	47 17 233 127 239 139 199 163
11	2 216	127 191 241 5 83 109 107 179
17	6 312	211 61 227 239 181 79 73 193
19	16 351	137 101 223 41 67 131 61 47

RSC encoder

After the interleaver block, we have two input strings. The original input and the interleaved input. The next step is to pass the input bit by bit through an RSC encoder. From Figure 6.2, we can see that two RSC blocks are in parallel. These two are identical, and the layout of the blocks is shown in Figure 6.3. Three output bits are calculated from each RSC encoder. This means we have six output bits for each bit, namely $X, Y_0, Y_1, X', Y'_0, Y'_1$. An input bit is processed for the first k clocks. After k clocks, the switch is closed, and trellis termination is handled [ITU22]. Trellis termination consists of six clocks. For the first three, only the output of the RSC encoder fed with original bits is output, and for the last three, only the RSC encoder fed with interleaved input are output.

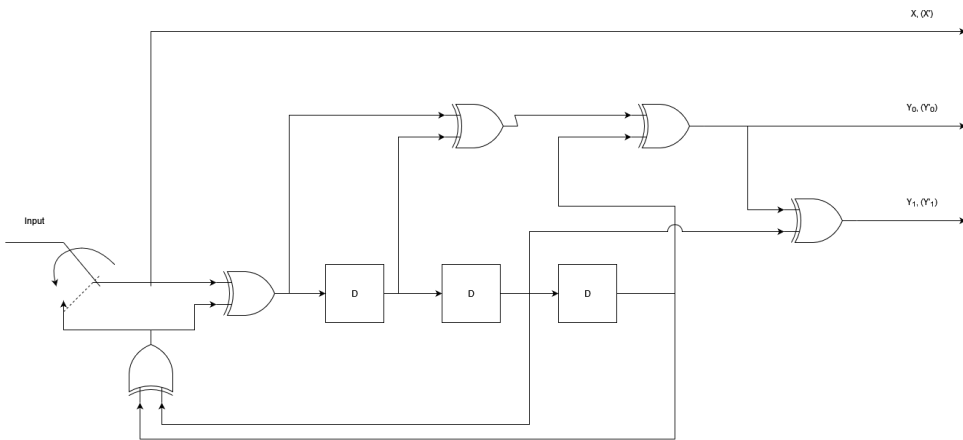


Figure 6.3: RSC encoder, (adapted from [ITU22])

From Figure 6.3, we discovered the operations needed to calculate the output bits associated with each input bit. The operations are elaborated in Equation (6.3). After the calculations, the registers are updated according to the figure before the next clock starts.

$$X = \textit{bit}$$

$$Y_0 = \textit{register}[2] \textit{xor} \textit{register}[1] \textit{xor} \textit{bit} \textit{xor} \textit{register}[0] \textit{xor} \textit{register}[2]$$

$$Y_1 = \textit{register}[2] \textit{xor} \textit{register}[1] \textit{xor} \textit{bit} \textit{xor} \textit{register}[0] \textit{xor} \textit{register}[2] \textit{xor} \textit{register}[1] \quad (6.3)$$

After k clocks, the operations change because there are no longer any processed input bits. The operations are shown in Equation (6.4) for the final output bits.

$$X = \textit{register}[1] \textit{xor} \textit{register}[2]$$

$$Y_0 = \textit{register}[2] \textit{xor} \textit{register}[1] \textit{xor} \textit{register}[0] \textit{xor} \textit{register}[2] \quad (6.4)$$

$$Y_1 = \textit{register}[2] \textit{xor} \textit{register}[1] \textit{xor} \textit{register}[0] \textit{xor} \textit{register}[2] \textit{xor} \textit{register}[1]$$

Puncturing

After each bit has been encoded into six output bits, puncturing is applied. For the first k clocks, the puncturing pattern is shown in Table 6.3. In the puncturing pattern, ‘1’ means that the bit shall be passed, and ‘0’ means that the bit shall be deleted. For example, if the output bits of the two RSC are “1,0,1,1,1,0”, and the puncturing pattern is “1,0,1,0,0,0”, the appended output bits are 11. If multiple bits are output, the output should be formatted in the order of $X, Y_0, Y_1, X', Y'_0, Y'_1$ [ITU22]. The puncturing pattern for each Link ID should be read from left to right and loop around after the last puncturing pattern has been applied.

Table 6.3: Puncturing pattern for some Link IDs , (adapted from [ITU22])

LinkID	Punc. pattern (X; Y0; Y1; X'; Y'0; Y'1 X; Y0; Y1; X'; Y'0; Y'1 ...)
5	1;0;1;0;0;0 1;0;0;0;0;0 1;0;0;0;0;0 1;0;0;0;0;0 1;0;0;0;0;0 1;0;0;0;0;1
11	1;1;0;0;0;0 1;0;0;0;1;0
17	1;1;0;0;0;0 1;0;0;0;1;0
19	1;0;1;0;0;0 1;0;0;0;0;0 1;0;0;0;0;0 1;0;0;0;0;0 1;0;0;0;0;0 1;0;0;0;0;1

After k clocks, the puncturing pattern changes. The puncturing pattern is shown in Table 6.4 for the final tail bit periods. For the first three clocks, only the output

from encoder one is output; for the last three clocks, only the output from encoder two is output [ITU22].

Table 6.4: Puncturing pattern for tail bit periods , (adapted from [ITU22])

Link ID	Punc. pattern tail bits (X; Y0; Y1; X'; Y'0; Y'1 X; Y0; Y1; X'; Y'0; Y'1 ...)
5	1;0;1;0;0;0 1;0;1;0;0;0 1;0;0;0;0;0 0;0;0;1;0;1 0;0;0;1;0;1 0;0;0;1;0;0
11	1;1;0;0;0;0 1;1;0;0;0;0 1;0;0;0;0;0 0;0;0;1;1;0 0;0;0;1;1;0 0;0;0;1;0;0
17	1;1;0;0;0;0 1;1;0;0;0;0 1;0;0;0;0;0 0;0;0;1;1;0 0;0;0;1;1;0 0;0;0;1;0;0
19	1;0;1;0;0;0 1;0;1;0;0;0 1;0;1;0;0;0 0;0;0;1;0;1 0;0;0;1;0;1 0;0;0;1;0;1

6.1.5 Bit Scrambling

Bit scrambling is required to prevent the power spectral density from being concentrated in the narrow band [ITU22]. The bit scrambler uses the polynomial shown in Equation (6.5), with the initialization sequence shown in Equation (6.6). After transmitting a packet, the bit scrambler is re-initialized with the initialization sequence. Our bit scrambling code was adapted from ⁵

$$F(x) = 1 + x^{14} + x^{15} \quad (6.5)$$

$$100101010000000 \quad (6.6)$$

6.1.6 Apply Syncword and Link ID

After bit scrambling the datagram, the next step is to apply the syncword and Link ID code word. For VDE-TER, the syncword sequence is shown in Equation (6.7).

$$111111001101010000011001010 \quad (6.7)$$

⁵<https://stackoverflow.com/questions/61229326/is-my-bit-scrambler-implementation-flawed>

The Link ID consists of six bits that are encoded into a sequence of 32 bits using a biorthogonal code [ITU22]. Some Link ID identification code words are shown in Table 6.5

Table 6.5: Link ID code words, (adapted from [ITU22])

Link ID	Bit scrambled code word
5	11 01 01 01 11 10 11 01 01 11 11 10 10 11 11 11
11	11 10 11 01 00 10 11 10 11 00 00 10 01 11 11 00
17	10 00 01 11 00 11 01 11 00 10 01 00 11 10 01 01
19	10 00 11 11 01 00 10 00 00 10 01 00 00 01 10 10

After the syncwords sequence and link identification code word have been identified, they are applied to the original datagram in the order shown in Equation (6.8)

$$\text{syncword} + \text{linkID_Code_Word} + \text{bit_scrambled_datagram} \quad (6.8)$$

6.1.7 Byte packing

Because GNU radio blocks for modulation only accept bytes, the next step is to byte pack the string of bits using Most Significant Bit (MSB) format.

6.1.8 Modulation

After the frame has been built, the next step is to modulate the signal according to the technical specifications. While AIS always uses GMSK as the modulation scheme, the modulation scheme for VDE-TER might vary depending on the Link ID. The syncword and Link ID might also be modulated with different modulation schemes to make it even more challenging.

Table 6.6: Modulation schemes for different Link IDs, (adapted from [ITU22])

Link ID	fragment of datagram	modulation
5	syncword	$\pi/4$ -QPSK (00/11 only)
	Link ID codeword	$\pi/4$ -QPSK
	remaining fragment	$\pi/4$ -QPSK

Continued on next page

Table 6.6: Modulation schemes for different Link IDs, (adapted from [ITU22]) (Continued)

Link ID	fragment of datagram	modulation
11	syncword Link ID codeword remaining fragment	$\pi/4$ -QPSK (00/11 only) $\pi/4$ -QPSK $\pi/4$ -QPSK
17	syncword Link ID codeword remaining fragment	$\pi/4$ -QPSK (00/11 only) $\pi/4$ -QPSK $\pi/4$ -QPSK
19	syncword Link ID codeword remaining fragment	$\pi/4$ -QPSK (00/11 only) $\pi/4$ -QPSK 16-QAM

From Table 6.6, we can see that Link IDs 5, 11, and 17 use the same modulation scheme. In VDES, the first output from the bit scrambler is mapped to the most significant bit until the least significant bit. VDES also uses an alternating $\pi/4 - QPSK$ mapping as shown in Figure 6.4.

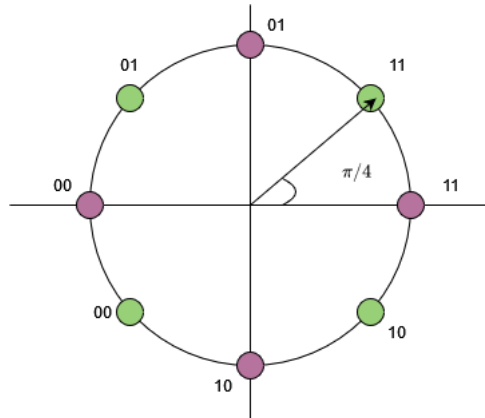


Figure 6.4: alternating $\pi/4 - QPSK$ mapping, (inspired by [ITU22])

The initial state of the $\pi/4 - QPSK$ mapping is defined by the green points shown in Figure 6.4, namely $(1 + j)/\sqrt{2}$, $(-1 + j)/\sqrt{2}$, $(-1 - j)/\sqrt{2}$, $(1 - j)/\sqrt{2}$. The next symbol is then mapped to the purple points, namely $(1 + 0j)$, $(0 + j)$, $(-1 + 0j)$, $(0 - j)$ [ITU22]. The third symbol is then mapped to the green points, the fourth to the purple, and so on. If the modulation of the remaining fragment is $\pi/4 - QPSK$, then the first symbol should be mapped to purple [ITU22].


```

correct FEC encoding: 00101000000000010110101111011110001100100001011101001110101111011110101
101000001010100001010000000010000010100000110100100010000010000000000000101000000100000010000
0000000000100000000000000000000010000010100000010000010000000101000001000000010000000101000000
0100000010000000000001010000000000010000001010000010100000010000010011011011100010100000110
110110011010110011010001

calculated FEC encoding: 00101000000000010110101111011110001100100001011101001110101011110111100
1011010000010101000010100000000100000101000001101001000100000100000000000001010000001000000010
0000000000001000000000000000000000100000101000000010000010000000101000001000000010000000101000
00001000000010000000000001010000000000001000000010100000010000000100000100110110111000101000000
110110110011010110011010001

Correct == calculated: True

```

Figure 6.7: Correct FEC encoding

$$\begin{aligned}
&0010100000000000101101011110111100011001000010 \\
&11101001110101011110111100101101000001010100001 \\
&01000000000100000101000001101001000100000100000 \\
&0000000000101000000010000000100000000000000100 \\
&00000000000000000000000010000010100000001000001000 \\
&00001010000010000000100000001010000000100000001 \\
&00000000000001010000000000000100000001010000010 \\
&10000000100000100110110111000101000000110110110 \\
&011010110011010001
\end{aligned} \tag{6.12}$$

Bit scrambling

The scrambled data is calculated by passing the output from the FEC encoder Equation (6.12) into the *bitscrambling()* function. The intended output is shown in Equation (6.13). Our result can be seen in Figure 6.8

```

correct scrambled: 001010111110110101110111011101111000011101011100000011000000101110011001
010111011111011010110011011010001110101101010001101111100111100000001011100100001011011011100000
101101011100001101111110001001000000111011010110000100101111001001010110000110011100111001100110
100110110101101101011101110111111100011010010001010001101001011000111010101101100100100001001
100110111110010101111

Calculated scrambled: 001010111110110101110111011101111000011101011100000011000000101110011
0010101111011111011010110011011010001110101101010001101111100111100000001011100100001011011011100
0001011010111000011101111110001001000000111011011000010010111100100101110000110011100111001100
1101001101101011011011110111111100011010010001010001101001011000111010101101100100100001
00110011011110010101111

Calculated scrambled == correct scrambled: True

```

Figure 6.8: Result bit scrambling

$$\begin{aligned}
&001010111110110101110111011101111000011110 \\
&10111000000111000000101110011001010111101111101 \\
&10101100110110100011101011010100011011111100111 \\
&10000000101110010000101101101110000010110101110 \\
&00011101111111000100100000011101101011000010010 \\
&11110010010101100001100111001110011001101001101 \\
&1010110110101110111011011111100011010010001010 \\
&00110100101100011101010101101100100100001001100 \\
&110111110010101111
\end{aligned} \tag{6.13}$$

Symbol mapping

After bit scrambling, the frame is finished by appending the syncword and Link ID codeword at the start of the packet as explained in Section 6.1.8. In Table 6.6, the modulation schemes for the different Link IDs are provided. This section will look at Link IDs 5, 11, and 17, as they all share the same modulation schemes. The syncword is modulated slightly differently than the rest of the frame. This is because '1' in the syncword sequence maps to '11', and '0' maps to '00', while two and two bits are read in for the rest of the frame. To ensure the syncword is read similarly to the rest, an easy cheat is to double each bit of the syncword before mapping. This means that '1' in the syncword sequence is doubled to '11', and '0' becomes '00'. However, since VDES uses an alternating $\pi/4 - QPSK$ modulation, we still need to make some changes to ensure the symbol mapping is computed correctly. By adding a third bit to every second bit, a zero if the green points are used, and one of the purple bits are used, we can set up our own constellation points with the correct mapping.

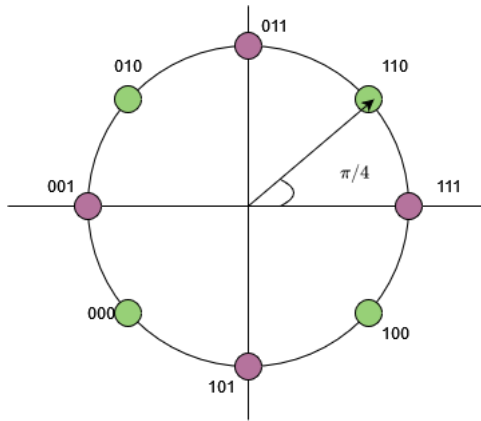


Figure 6.9: Extending the modulation scheme

Using the new constellation points, we can first read three and three bits, the most significant bit, and map them to the appropriate constellation points. From the ITU specification, we know that the symbol mapping for the output message (syncword sequence + Link ID codeword + Equation (6.13)) is Table 6.7

Table 6.7: Correct symbol mapping from ITU, [ITU22]

Syncword sequence symbols
(+0.7,+0.7),(+1.0,+0.0),(+0.7,+0.7),(+1.0,+0.0),(+0.7,+0.7),(+1.0,+0.0),(-0.7,-0.7),(-1.0,+0.0),(+0.7,+0.7),(+1.0,+0.0),(-0.7,-0.7),(+1.0,+0.0),(-0.7,-0.7),(+1.0,+0.0),(-0.7,-0.7),(-1.0,+0.0),(-0.7,-0.7),(-1.0,+0.0),(-0.7,-0.7),(+1.0,+0.0),(+0.7,+0.7),(-1.0,+0.0),(-0.7,-0.7),(+1.0,+0.0),(-0.7,-0.7),(+1.0,+0.0),(-0.7,-0.7)
Link ID code word symbols
(+1.0,+0.0),(-0.7,+0.7),(+0.0,+1.0),(-0.7,+0.7),(+1.0,+0.0),(+0.7,-0.7),(+1.0,+0.0),(-0.7,+0.7),(+0.0,+1.0),(+0.7,+0.7),(+1.0,+0.0),(+0.7,0.7),(+0.0,-1.0),(+0.7,+0.7),(+1.0,+0.0),(+0.7,+0.7),
Bit scrambled symbols
(-1.0,+0.0),(+0.7,-0.7),(+0.0,-1.0),(+0.7,+0.7),(+1.0,+0.0),(+0.7,+0.7),(+0.0,+1.0),(+0.7,0.7),.....,(+1.0,+0.0),(-0.7,+0.7),(+1.0,+0.0),(+0.7,+0.7),(-1.0,+0.0),(+0.7,-0.7),(+0.0,-1.0),(+0.7,+0.7),(+1.0,+0.0)

Table 6.8: Result of our Symbol mapping

Syncword sequence symbols
(0.7,0.7),(1,0),(0.7,0.7),(1,0),(0.7,0.7),(1,0),(-0.7,-0.7),(-1,0),(0.7,0.7),(1,0),(-0.7,-0.7),(1,0),(-0.7,-0.7),(1,0),(-0.7,-0.7),(-1,0),(-0.7,-0.7),(-1,0),(-0.7,-0.7),(1,0),(0.7,0.7),(-1,0),(-0.7,-0.7),(1,0),(-0.7,-0.7),(1,0),(-0.7,-0.7),
Link ID code word symbols
(1,0),(-0.7,0.7),(0,1),(-0.7,0.7),(1,0),(0.7,-0.7),(1,0),(-0.7,0.7),(0,1),(0.7,0.7),(1,0),(0.7,-0.7),(0,-1),(0.7,0.7),(1,0),(0.7,0.7)
Bit scrambled symbols
(-1,0),(0.7,-0.7),(0,-1),(0.7,0.7),(1,0),(0.7,0.7),(0,1)(-0.7,0.7),.....,(1,0),(-0.7,-0.7),(1,0),(-0.7,0.7),(1,0),(0.7,0.7),(-1,0),(0.7,-0.7),(0,-1),(0.7,0.7),(1,0)

From Table 6.8, we can see that our symbol mapping was correct and is the same as in Table 6.7. By doing this step, we can byte pack the string of bits and send them to a constellation block, where our custom constellation points are implemented.

Validity of Results

Since the only example included in the ITU technical specification for VDES is for ASM, there might be some differences regarding VDE-TER that we have not identified. One possible error is regarding the endianness in VDE-TER. In the Annex regarding ASM messages, endianness is not mentioned, but for VDE-TER, it is mentioned that the same endianness is used as in AIS [ITU22]. The endianness scheme for AIS is shown in Figure 5.7, and is performed on the data after the CRC-16 bit sequence has been appended.

Since no examples of burst generation for VDE-TER are appended in the technical specification for VDES, we cannot verify at what stage of the frame generation endianness is performed. We, therefore, decided to implement endianness at the same stage as in the AIS frame builder, namely after the CRC-32 check sequence has been appended. The bits were also reversed for each byte in the syncword and the Link ID codeword, similar to the preamble, start flag, and end flag in AIS.

This was implemented by applying the *revers_bit_order()* function from the ais frame builder and passing the *payload_crc* parameter from the VDE-TER frame builder as input. The reverse bit order for each syncword and Link ID codeword was hard coded. This was done because the syncword consists of 27 bits, which means that if endianness were performed on the entire string, the syncword bits and Link ID bits would be intermixed. However, we have no way to validate that endianness

is performed correctly. When exemplified VDE-TER messages are available in the future, this can be verified or disproven. We believe the structure of the VDE-TER frame builder is well explained, and changing possible mistakes should be possible.

6.2 Building a GNU Radio Signal Simulator for lower leg VDE-TER channel 1024, with Link ID 11

After building a frame builder for VDE-TER messages, the next step is to build a signal simulator for VDE-TER in GNU Radio. Following the architecture of the original AIS_TX built by Balduzzi et al., we tried to implement a similar system with blocks and parameters intended for VDE-TER. From AIS_TX, two parameters stand out. Namely the bit rate and sampling rate. For AIS, the bandwidth time product and bit rate are included in the technical specification, respectively 0.4 and 9.6 kbit/s [ITU14].

The bit rate for VDE-TER is not specified as it might vary depending on the Link ID. However, the symbol rate is included. The symbol rate for Link ID 11, 17, and 19 is 19.2, 38.4, and 76.8, respectively. Link ID 11 and 17 use $\pi/4 - QPSK$ as their main modulation. The bit rate for Link ID 11 and 17 is, therefore, symbol rate * 2, namely 38.4 and 76.8. However, since we appended one extra bit for every second bit, our bitrate is $3/2 * bit_rate = 57.6$. We can calculate the sampling rate from the bit rate and symbol rate, which is $bit_rate * symbol_rate$.

In the AIS frame builder developed by Balduzzi et al., the amplitude range of the signal is reduced from $+ - 1$ to $+ - 0.9$ to prevent signal clipping of the SDR peripheral [BPW14]. We, therefore choose the same parameter. Only channel 1024 was implemented to simplify the signal simulator as much as possible. This means that only a bandwidth of 25 kHz is available, and therefore only Link ID 11 from VDE-TER can be used. In Figure 6.10, the final signal simulator, which might be able to transfer VDE-TER messages over lower leg TDMA channel 1024, is provided. The grc file is included in [LB23]⁶.

Sending 6-bit ASCII encoded text

To portray how the frame builder and exemplified signal simulator work, we identified the necessary steps to send a short data message containing 6-bit ASCII-encoded text. As shown in table 6.1, the short data message contains a payload field, which is not defined. The payload field was therefore filled with the VDE protocol format defined for applications that use VDE to transmit text using 6-bit ASCII [IAL22]. We cannot verify if this is correct, but we determined to add the specified protocol

⁶<https://github.com/collinlokken/grc-ais-vdes-framebuilder/blob/main/vdetx1024.grc>

6.2. BUILDING A GNU RADIO SIGNAL SIMULATOR FOR LOWER LEG VDE-TER CHANNEL 1024, WITH LINK ID 11 75

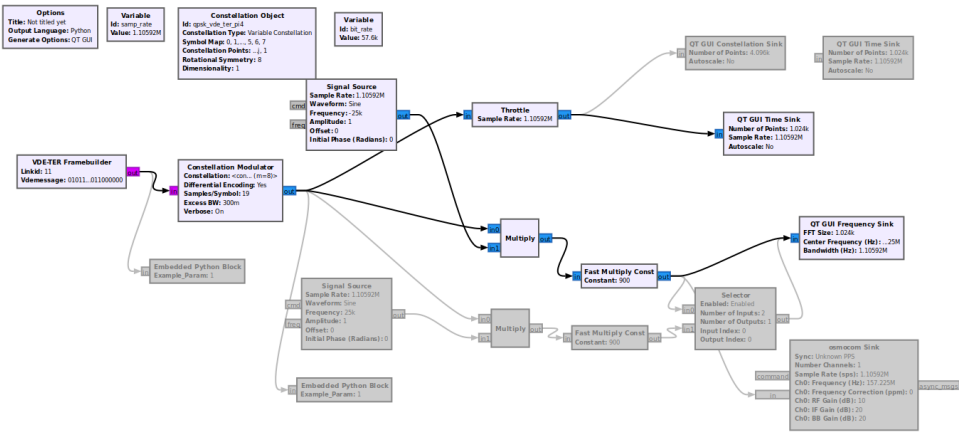


Figure 6.10: A signal simulator for upper leg channel 1024, with Link ID 11

format in the variable payload field as no other information was available. The protocol format is shown in Table 6.9

Table 6.9: Protocol Format for 6-bit ASCII text, adapted from [IAL22])

Parameter	Size in Bits	Description
VPMI	16	2
Message ID	16	0
Require Acknowledge	1	1 if ACK is required, 0 otherwise
Sequence Number	2	0-3, number of retries
Text sequence number ID	11	Sequence number to be incremented by application, 0 indicates no sequence numbers are used.
Text String	6-6000	6-bit ASCII encoded text
Spare bits	0-7	0 padding to ensure bytes are filled up
Total number of data bits	variable	Total number of bits in the protocol format

The program used to generate the payload script was written in Python and can

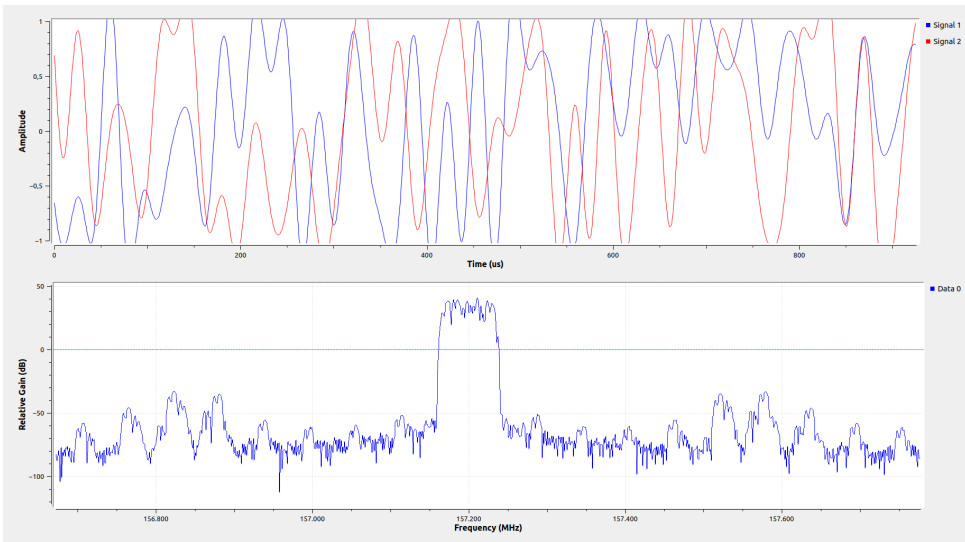


Figure 6.11: Results from Frequency Sink and Time Sink for short data messages

Chapter 7

Analysis

This chapter presents a comprehensive analysis of spoofing AIS messages following the introduction of VDES technology. Additionally, we explored the potential vulnerabilities of services that may be provided via VDES using STRIDE.

To assess the financial implications of executing such attacks, we employed the RCM framework developed by Haga [Hag20]. By synthesizing the findings from our extensive literature review, interviews, and experiments, we could ascertain the costs associated with each stage of these attacks. This examination enabled us to understand better the feasibility and potential repercussions of such malicious activities in a VDES-enabled environment.

7.1 Analysis AIS

This section aims to build on the work by Walde et al., where they explored the resources necessary to carry out attacks on AIS [WH20]. This section examines the cost of conducting similar attacks on AIS 2.0, VDES. In a CPA attack, an adversary introduces a ghost ship by forging AIS messages and transmits them to the intended target. At the receiver end, the ECDIS shows that the ship is heading towards a collision and must change its course. The result of the attack might be that the target strays off course and travels into dangerous waters. we discovered three different scenarios, shown in Section 7.1. From the Interviews discussed in Chapter 4, it was stated that even though VDES has the possibility of authenticating AIS, it is likely, not mandatory. In the overview of VDES from IALA, an exemplified protocol that transfers positional AIS data was included [IAL22]. We, therefore, decided to cover three different scenarios, shown in Section 7.1.

– Scenarios

Scenario 1: A PKI is available, but optional.

Scenario 2: The target requires a Signature

Scenario 3: VDE-SAT enables retransmitting of AIS messages.

7.1.1 AIS: Scenario 1, A PKI is available but optional

Assumptions

Because VDES needs to be compatible with legacy AIS systems, the AIS functionality of VDES must remain the same. As proven in the experiment, it is still possible to spoof AIS on the new VDES system. However, IALA has identified a VDE protocol format for transmitting a digital signature for a previous AIS message over VDE-TER. The VDE-TER message containing the signature might be used to augment received AIS messages on both shore and ship [IAL22]. The protocol format is included in Table 7.1. In the technical specifications for VDES, it is stated that it is assumed that a PKI is established with an international organization, capable of acting as a CA [ITU22]. In Scenario 1, authentication of AIS messages is possible but not mandatory. An attacker can therefore force the target into receiving a false AIS message, perceiving it as true, without providing a valid signature.

Table 7.1: Protocol format of signed AIS message (adapted from [IAL22])

Parameter	Number of bits	Description
VPFI	16	2
Message ID	16	6
AIS message ID	6	AIS message, 1 upto 27
MMSI	30	Unique ship identifier
Channel ID	2	channel of the original transmission 0: AIS 1 1: AIS 2
Slot number	12	Slot number value 0 .. 2249, 2250 if unspecified
Timestamp	32	Included to avoid replay attacks. Should be set to the time when the AIS message was transmitted

Continued on next page

Table 7.1: Protocol format of signed AIS message (adapted from [IAL22]) (Continued)

Parameter	Number of bits	Description
Signature	512 bits	Signature of all the data above and the ais message to be signed. String_to_sign = this_message + ais_message.

Reconnaissance

The first stage of the CKC is the reconnaissance stage. In the reconnaissance stage, the attacker gathers information about their target and explores potential vulnerabilities. The reconnaissance is shown in Figure 7.1 and is inspired by the work conducted by Walde and Hanus [WH20]. Marine traffic can be used to determine the target vessel's location. Marine traffic is free, but a global satellite plan that allows tracking of all ships, costing \$312¹ exists. An adversary could also buy a commercial VDES receiver or set up his own AIS receiver as explained in Chapter 5 and use this to gather the necessary information about the target. Considering the cost of a HackRF One is, \$360². Since some work is needed to install all the required software, the total cost is around \$400. We also contacted a company that states they have made a SDR solution for VDES. The cost of the device is \$4000.

It is difficult to estimate the cost of a bribe. We would still argue that a cost of \$5000 would challenge the moral compass of most moral people. Still, our confidence regarding this resource alternative is low as we have no data regarding this. As sending e-mail is free, the cost of phishing or other social engineering methods is set to \$0.

¹<https://www.marinetraffic.com/en/online-services/plans>

²<https://www.amazon.com/s?k=hackrf>

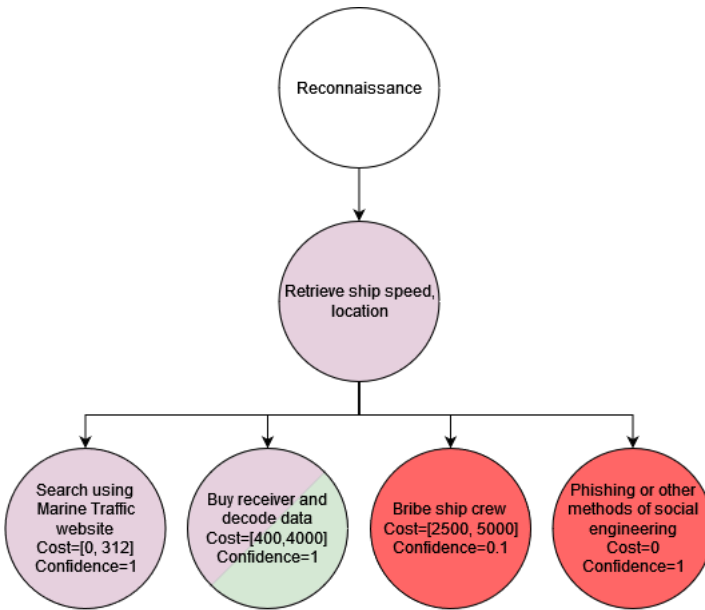


Figure 7.1: Reconnaissance stage, adapted from [WH20]

Weaponization

The second stage of the RCM is the Weaponization phase. The attacker needs a signal simulator to forge AIS messages at this stage. Since VDES needs to be compatible with AIS, signal simulators for AIS can still be used to simulate the signal. An option is, therefore, to use the open-source Software AIS_TX. AIS_TX is, however, old and originally written for GNU Radio 3.7. An option now is to use the AIS frame builder, implemented in Chapter 5. Since our AIS frame builder is implemented as a Python block, the installation step is simplified considerably for GNU Radio 3.8 and upwards. We also consider our AIS frame builder easier to use as the parameters can be set inside GNU Radio, and the frame builder does not need a valid AIS message as input. The cost for each step is estimated by the formula $\$20\text{cost}/\text{hour} * \text{hours of work}$, and the hours of work are based on our experience from the experiment Chapter 5.

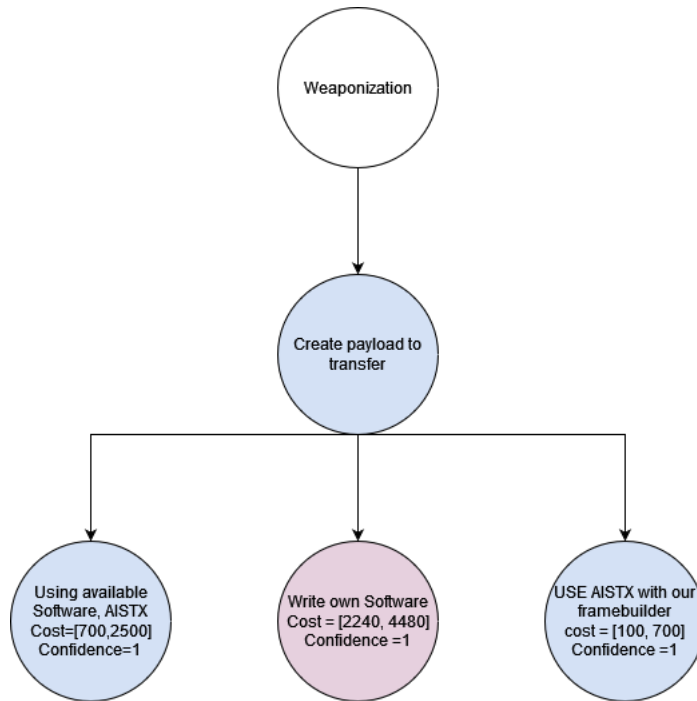


Figure 7.2: Weaponization stage, AIS: Scenario 1, adapted from [WH20]

Delivery

In the delivery stage of an attack, the weaponized payload is delivered to the victim. Several resources are needed to deliver the AIS payload to the victim. First, the adversary must acquire the necessary hardware to transfer AIS messages. The experiment showed how an attacker could use an SDR device such as the HackRF One to transmit AIS messages. The HackRF One is available online and priced at roughly \$359³. Another option is to use the more expensive USRP, which is priced at roughly \$1600⁴. In our experiment, we transmitted the messages through a wire, an adversary, however, would most likely need an antenna optimized for AIS signals. Such devices are priced anywhere from \$79 to \$200⁵. Then the adversary must download the necessary drivers and programs to use the available hardware and open-source software.

Finally, to deliver the weaponized payload, the adversary must be in the transmission range of the target. An adversary can do this by traveling in a boat or transmitting from the coast if the ship is close to shore. To easier compare our results

³<https://www.amazon.com/s?k=hackrf>

⁴<https://www.ettus.com/all-products/usrp-b200mini-i-2/>

⁵https://www.milltechmarine.com/AISVHF-Antennas_c_55.html

with Walde and Hanus, we opted to use the same cost they set for this resource alternative, namely [\$200, \$5000 with a confidence level of 0.1 [WH20].

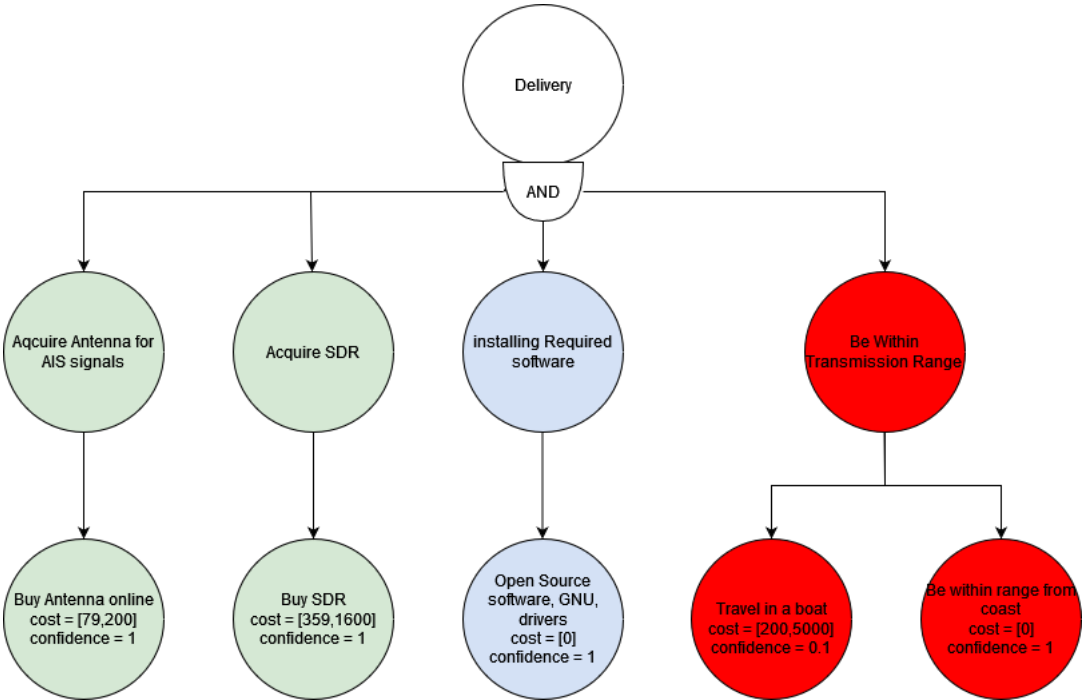


Figure 7.3: Delivery stage, AIS: Scenario 1, adapted from [WH20]

Exploitation

In the exploitation stage, the attacker utilizes that no authentication nor encryption is provided in the AIS messages.

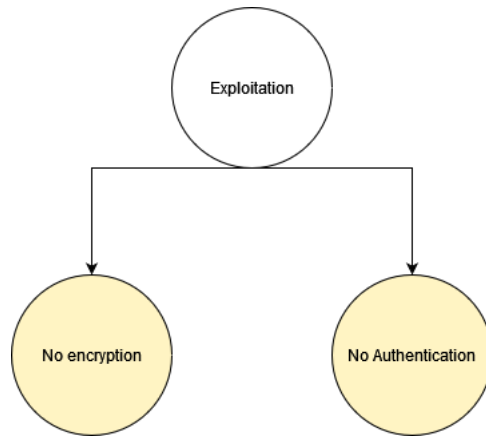


Figure 7.4: Exploitation stage, AIS: Scenario 1, adapted from [WH20]

Command and Control

Sending a single AIS message is insufficient to maintain control over the target. Depending on the transponder class, speed, and rate, the AIS transmission rate might vary between every two seconds and every 3 minutes [ITU14]. In a CPA attack, the most likely scenario is that the adversary introduces a ghost ship that travels quickly towards the CPA to stress the target. In this case, the vessel's position is transmitted every 2 seconds. This is faster than a user can generate, copy and print the binary message into AIS_TX. An adversary must use a program that takes a new AIS message as input and transmits it every two seconds. Such a program can be combined with AIS_TX but has to be written. Alternatively, the attacker can use our AIS frame builder to update the AIS messages in real time, ensuring command and control. The command and control stage is shown in Figure 7.5.

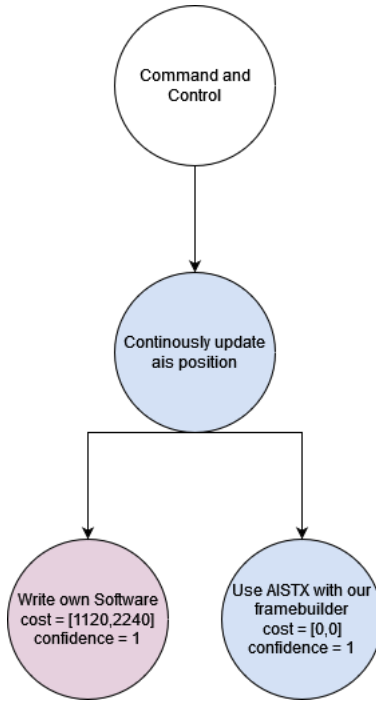


Figure 7.5: Command and Control, AIS: Scenario 1

Actions on Objective

For this attack to work, it is essential that the ghost ship does not conduct any maneuvers outside the laws of physics. This is because if the ghost ship AIS messages are suspicious, the target would most likely understand that it is dealing with a ghost ship. It is, therefore, important that the parameters in the AIS message are adjusted according to the laws of physics.

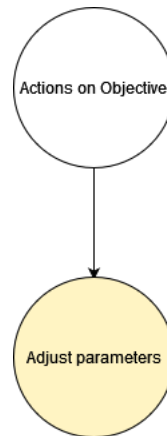


Figure 7.6: Actions on Objectives stage, AIS: Scenario 1, adapted from [WH20]

Cost Estimation

The total cost and confidence of the attack is shown in Table 7.2

Table 7.2: Estimation of Cost

Stage	Min Cost	Max Cost	Conf
Reconnaissance	0	5000	0.76
Weaponization	100	4480	1
Delivery	438	6800	0.55
Exploitation	0	0	1
Installation	Left Out	Left Out	Left Out
Command & Control	0	2240	1
Actions On Objective	0	0	1
Total	538	18520	0.418

7.1.2 Scenario 2: (CPA) with mandatory signature

Assumptions

If the receiving ship demands a signature of the AIS message, an attacker would need access to the private key associated with the ghost ship. For this attack to work, it is also necessary that the target is outside a control station service area and the signature is transferred on a ship-originated broadcast. This is because if the ship-to-ship direct message is used, the receiver returns a resource allocation message. In this case, the adversary would need a transmitter and a VDE-TER receiver to

determine the frequency, link ID, and time it should transfer on. Of course, the attacker could try to brute force this by continuously transferring the message on all the TDMA channels and changing the link ID, but this is likely not to work, as he might transfer on the right channel but with the wrong Link ID at the right time.

Reconnaissance

The reconnaissance step in Scenario 2 is similar to the stage described under Section 7.1.1. It is, therefore, not elaborated upon further here.

Weaponization

To generate forged AIS messages and VDE-TER, both an AIS and VDE-TER signal simulator is necessary. AIS messages can be generated using the open software tool, AIS_TX, or using the AIS_TX architecture with our AIS frame builder. We have been unable to locate any open-source software for VDE specific messages. This is likely because the VDES is still in the testing phase and is not commercially available. In Chapter 6, we have explained how an attacker could possibly build a VDE-TER signal simulator to send short data messages over VDE-TER. However, our confidence regarding its functionality and reliability is low due to the lack of testing on the signal simulator.

If an adversary decides to use our signal simulator, he would first have to verify that the signal simulator works as intended. Then the adversary would have to identify what channel the signature should be transmitted over the corresponding link ID and write a VDE message generator program that generates the correct message. It is hard to estimate the cost for all these steps, so the confidence level is low. We estimate an attacker would have to spend anything between two and four workweeks to accomplish this. We also contacted a company that states they have made their own SDR for VDES. The cost associated with this device is \$4000. The weaponization is shown in Figure 7.7.

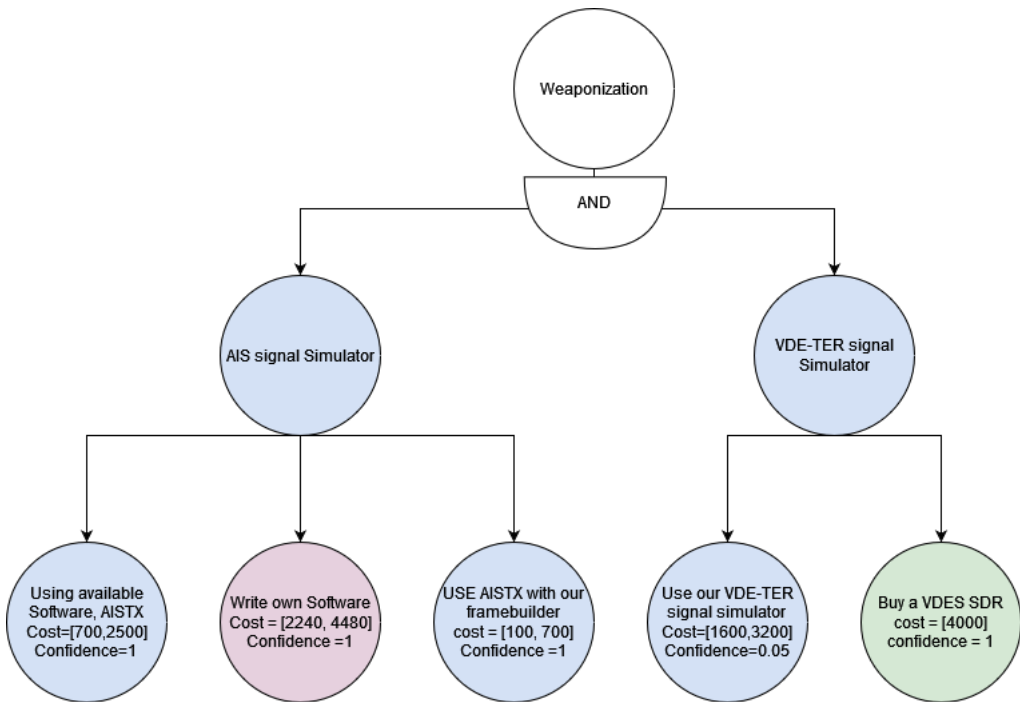


Figure 7.7: Weaponization stage, AIS: Scenario 2

Delivery

This step did not differ from Scenario 1; see Section 7.1.1.

Exploitation

For this attack to work, the attacker needs to have obtained a valid private key. With a valid private key, an adversary can provide a valid signature for the forged AIS message. Since the time stamp is included in the digital signature, it is impossible to record an original authenticated AIS message and replay it later. The adversary would need access to an official private key to fake an authenticated AIS message. Different approaches exist but are costly for ECDSA, with a public key length of 256 bits. With a key size of 256, the use of SHA-256, and the elliptic curve, secp256r1, it is stated that the minimum security bits are 128 bits [ITU22]. 128 bits of security is often considered sufficient for most applications [Bar20].

In other words, finding the private key from the public key with a minimum of 128 security bits is considered unfeasible. Therefore, an attacker would need a different approach. While not specific to ECDSA, an attacker could still try to manipulate individuals into revealing their private keys through social engineering, phishing,

or bribing. It is hard to estimate the cost of bribing a crew member into revealing information, as the cost might vary from person to person and each individual’s life standard and morals.

Another approach is to break onto a boat and steal the unit holding the private key, for example, the smart card. Depending on the security of the separate unit, the unit might be easy or hard to exploit. If the unit is password protected, the adversary might need the password to use it to sign messages. We did not research this and cannot come up with an exact cost, but we think this is an interesting topic for future research. After stealing the unit, the adversary would only have a short window of action, as the theft would be reported, and the certificate for the key would be revoked. Thus, if a smart card is reported stolen and available online, but a ship has not updated its entity, an adversary can still use it to forge valid signatures. Another approach for an adversary is to obtain a private key by bribing an already vetted organization registered in an MCP instance to register a new fraudulent ship. The entire exploitation stage is shown in Figure 7.8.

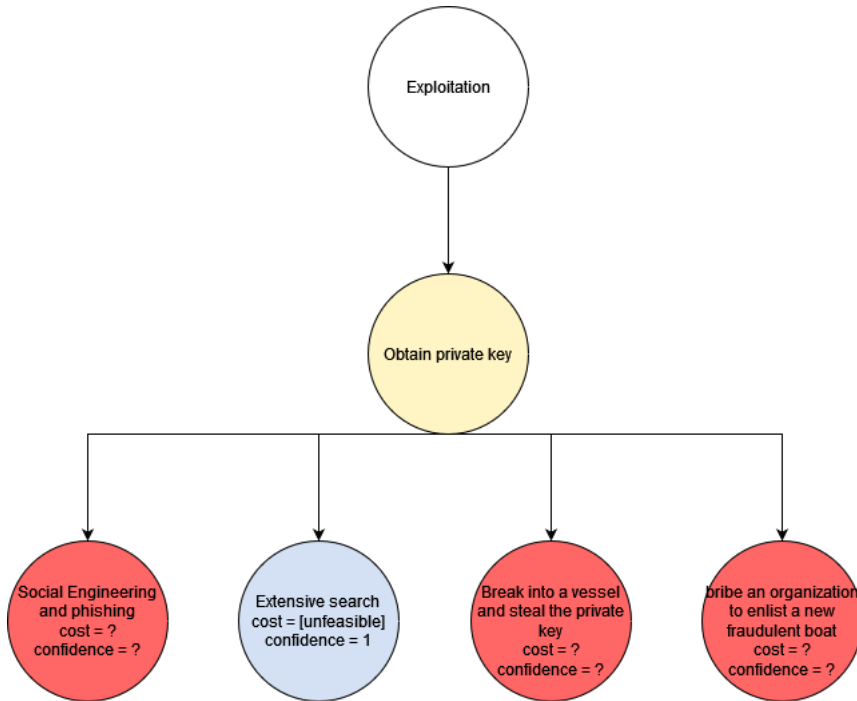


Figure 7.8: Exploitation stage, AIS: Scenario 2

Command and Control

This step did not differ from Scenario 1; see Section 7.1.1

Actions on Objective

This step did not differ from Scenario 1; see Section 7.1.1.

Cost estimation

The total cost and confidence of the attack is shown in Table 7.3

Table 7.3: Total Estimation of Cost

Stage	Min Cost	Max Cost	Conf
Reconnaissance	0	5000	0.76
Weaponization	1700	8480	0.53
Delivery	438	6800	0.55
Exploitation	∞	∞	1
Installation	Left Out	Left Out.	Left Out
Command & Control	0	2240	1
Actions On Objective	0	0	1
Total	∞	∞	0.22

7.1.3 Scenario 3: Retransmitting AIS position reports over VDE-SAT

Assumptions

In our introduction, we speculated that introducing VDES would make it easier for adversaries by reducing the required resources necessary for spoofing AIS messages. A complete overview of VDES functions and frequencies is shown in Figure 2.2. This figure shows that ships can send AIS messages to satellites using AIS channels 1 and 2. However, no AIS channels for communication from satellites to ships exist. The only downlink channel from satellite to ship is the SAT2 Downlink channel. Therefore, to transmit AIS messages to ships beyond the AIS reception range, VDE satellites must transmit over VDE-SAT.

In the VDES documentation, an example VDE Protocol Format that optionally retransmits AIS messages over VDE-SAT is proposed. The VDE Protocol Format is included in Table 7.4. Only positional AIS messages with a repeat indicator of value 0 are included [IAL22]. The repeat indicator of value 0 indicates that the message origin is the original ship's message and has not been repeated by a base station. Legacy AIS repeater stations have a filtering mechanism, which is a configurable function, that filters messages that the repeater station should repeat. The filtering considers

message type, coverage area, and required message reporting interval [ITU14]. Iala states that the signature is only meant to authenticate the receiver's validity, not the data of the AIS messages [IAL22].

By utilizing the VDE-SAT downlink, ships beyond the range of the AIS reception range in remote areas gain better situational awareness. This feature of VDE could, however, also possibly enable adversaries out of range of a traditional AIS receiver to send unauthenticated AIS messages to ships that previously have been beyond the adversary's scope. An adversary can potentially reach all ships globally, depending on how the filtering mechanism works. With the proper hardware, an attacker might make a satellite repeat a forged AIS position message. It is not stated if the repeat indicator value is incremented at the satellite. Still, if we assume the satellite follows the legacy specification, the repeat indicator shows that the AIS message has been repeated. Depending on how the receiving ship interprets the message, an adversary might be able to conduct a CPA attack, tricking the receiving ship into dangerous waters.

Table 7.4: Retransmit AIS messages over VDE-SAT (adapted from [IAL22])

parameter	Number of bits	Description
VPMFI	16	2
Message ID	16	5
n	8	number of encapsulated AIS position reports 0: unused value 1-255: valid values
Time [0]	40	Time when the first AIS message was received
AIS message [0]	168	First AIS messages
...
Time [n-1]	40	Time the n-1 AIS messages was received
AIS message [n-1]	168	n-1 AIS message
Valid until	32	Should be set to the time when the message was transmitted to avoid replay attacks
Signature	512	Signature of all the data above. Generated as described in Section 2.2.5

A similar environment to Scenario 1 is needed for such an attack. Accordingly, all the steps of the Cyber Kill Chain are similar, except for the delivery stage. Therefore, only the delivery stage is included in this section. See Section 7.1.1 for the rest of the steps.

Delivery

In the previous descriptions of the delivery stage, the attacker must be within the target antenna's reception range. Depending on the target's location, the attacker might need access to a boat or be within range of the AIS receiver from the coast. However, by utilizing the VDE-SAT downlink, an attacker might gain easier access using a VDE satellite. Therefore, the need to be within transmission range of AIS might not be necessary. See Figure 7.9 for the delivery stage.

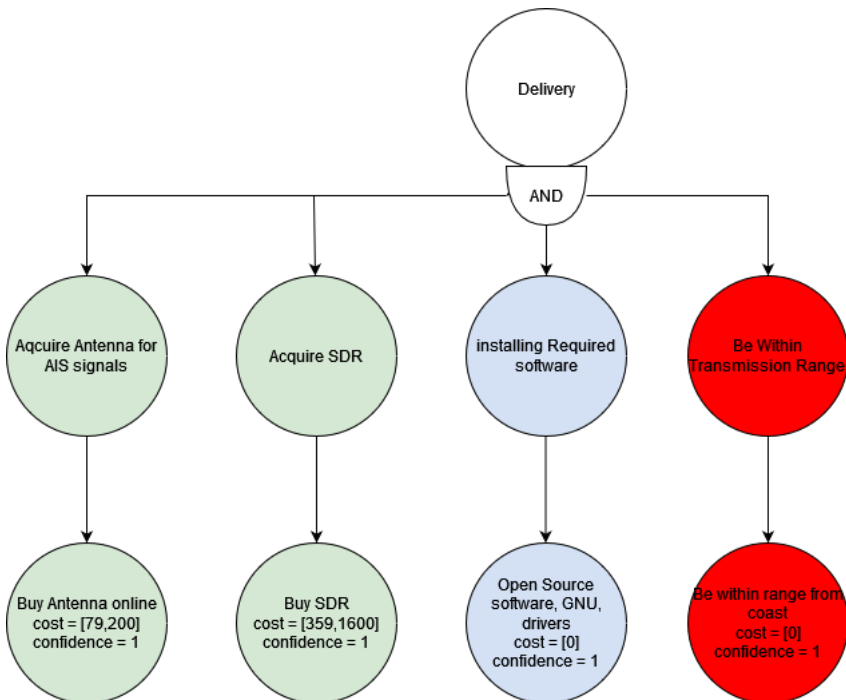


Figure 7.9: Delivery stage, AIS messages transmitted via VDE-SAT

Cost Estimation

The total cost and confidence of the attack are shown in Table 7.5

Table 7.5: Estimated cost for Scenario: 3

Stage	Min Cost	Max Cost	Conf
Reconnaissance	0	5000	0.76
Weaponization	100	4480	1
Delivery	438	1800	1
Exploitation	0	0	1
Intallation	Left Out	Left Out	Left Out
Command & Control	0	2240	1
Actions On Objective	0	0	1
Total	538	13520	0.76

7.2 Analysis VDES

This section introduces the main analysis of one future VDES-based service. It is, therefore, appropriate to utilize the S-100 standard as a foundation when we try to imagine the future services that can be offered over VDES and the challenges they face. Amongst the products listed in Table 2.3, we chose to look at the S-129 Under Keel Clearance Management (UKCM) data product specifically. UKCM was identified from our interviews as one of the data products with the highest potential and is well documented, which makes it appropriate for the analysis. However, many of the threats described also apply to other S-100 data products. Therefore, it is possible to extend the identified attacks to other S-100-compliant services as well.

To conduct the analysis, we used the frameworks specified in Chapter 3, to arrive at a cost estimate for potential attack vectors aimed at UKCM. Lastly, we conduct a comparative assessment to see if the cost changes when introducing VDES as a delivery mechanism. We limited the scope of our analysis to attacks that directly target the data exchange mechanism because we are looking at the security of VDES-based services. These are attacks where VDES plays an important role. First, we introduce the service and give an overview of how the service operates and the value gained by the end user. Then we look at potential misuse cases and consequences before identifying the threats that can trigger these consequences using the STRIDE methodology. Next, we used the RCM to analyze how an adversary can carry out the identified threats. This helped us to gain a better overall view of the feasibility of the attacks and associated costs.

7.2.1 Assumptions

As discussed in Section 2.4, there is still uncertainty related to how mechanisms such as delivery, discoverability, and authentication will be implemented in the future. The majority of international standards in this area are still in the draft stage, and many challenges related to security infrastructure, such as the certificate revocation problem in PKI[BBBM21], are yet to be solved. In the current environment, assuming that these challenges will be solved initially would not be appropriate, so we took a more conservative approach for the following analysis, and we assume that the optional SMMP is not utilized. In our scenario, the service providers deliver their services while assuming that the security implementations of S-57 and S-100 are sufficient to provide a cyber-secure service. No further authentication or encryption is provided on the messages transferred over VDE-SAT. An important consequence of this is that MMS agents can no longer verify a message's origins, which is something an adversary can exploit.

It is also unclear if end users transitioning to VDES can still use S-57 data products, as the standard is cryptographically outdated, as discussed in Section 2.3. The S-100 standard uses keys that are stronger but still theoretically possible to attack. Another possibility is that early adopters of VDES can possibly be forced to use S-57 while S-100 data products are being developed. Meaning that S-57 can possibly be sent over VDES while the industry is transitioning. The question becomes if a customer of VDES buys a VDES module before or after all the services are ready. Here one could argue that a popular S-100 data product might draw many new users to VDES if they want to gain access to the service early. In this case, all S-57-equivalent S-100-based data products might not be available, and some S-57 ENCs must be used.

It is also unclear if legacy systems can use S-100-based data products, as S-100 does not require VDES. However, legacy systems are not built to handle the cryptographic keys of S-100, so the ENC distributor would perhaps have to downgrade to shorter keys if it becomes relevant. UKCM is an S-100-based service, and because we are looking at VDES-based services, it is safe to assume that the end users in our scenario would have a VDES module on board, which is able to handle the cryptographic scheme presented in S-100.

7.2.2 Product Description: Under Keel Clearance

S-129 describes Under Keel Clearance as a NPIO which is responsible for encoding UKCM information from a UKCM service provider. According to SOLAS, the ship's master must plan the ship's voyage, which includes planning safe passage through waterways where the risk of grounding is high, also known as a UKCM operational area. The ship master sends updated information about the ship to a UKCM service

provider, who uses specialized mathematical modeling tools to predict if the area in question is navigable for the ship. The calculations are based on local weather, currents, and water levels for the time of passage.

The calculation results are further used to plan a safe route for the ship, which, approximately 24 hours before arriving at the UKCM area, is updated by the service provider based on more recent readings of the initial information to provide a more detailed plan. In case of an update to the original plan, the ship may manage its speed accordingly to arrive during the specified time window when the passage is safe. These updates must be consistent with the ship's actual speed, heading, and position, which the UKCM provider reads from AIS transmissions from the ship [IHO19].

7.2.3 Description of possible attacks

Attacks against UKCM services would attempt to disturb the view presented to the ship's master, leaving them with a view removed from reality. In the view, an area can be in three states, either non-navigable (high risk), almost non-navigable (medium risk), or navigable (low risk) [IHO19]. For simplicity, we chose to look at two states: high and low risk.

An attacker's job would be to modify these states so that the view presented by the navigational equipment shows a different risk than reality. If an area is shown as low risk when it is not, it can cause the ship to enter dangerous waters and ground. If the area is shown as high risk when it is not, it can cause the ship to take a different route to avoid the high-risk areas, potentially wasting many resources if the rerouting is costly. In the worst-case scenario, the ship cannot move if the passage is narrow, so the ship cannot turn around. One could also imagine trapping the ship by surrounding it with only high-risk areas.

The attacker can also indirectly modify the ship's view by denying updates from authorities, leaving the ship's master with an incomplete view of the ship's surroundings. In each scenario, the attacker aims to make the ship ground or waste resources. A successful attack could have grate economic consequences and loss of life in the worst case. Looking at these scenarios from the perspective of STRIDE, we have been able to identify the following Threat Consequences (TCs) for each threat category:

Spoofting

Spoofting areas of clearance can cause ships to ground if the ship thinks that the clearance is greater than it is (TC-1) or cause the ship to waste resources if it is less than it is (TC-2).

Tampering

Tampering with the AIS messages transmitted by the ships to the service provider can cause the service provider to calculate incorrect values if this does not match the ship's weight, speed, heading, position, etc.(TC-3). Tampering with sensor readings that monitor weather, water levels, currents, and other relevant metrics so that authorities and ships get a false view of the safety level of the area (TC-4).

Repudiation

Insufficient logging in the ECDIS can allow repudiation of received charts, routes, and updates (TC-5).

Information disclosure

Eavesdropping on the ship's planned route and route updates can allow pirates to plan an ambush on the ship (TC-6).

Denial of service

Jamming the reception of charts or chart updates reduces the ability of the ship to detect updates to dangerous areas (TC-7)

Elevation of privileges

The attacker can attempt to become a certified service provider, allowing them to further spoof or tamper with information sent to the ships (TC-8).

7.2.4 Attack: Exchange set forgery

Description

This attack aims to manipulate the victim's view, aiming to send the ship into dangerous situations. To some degree, the crew onboard the ship trusts the view presented by onboard navigational equipment. They might get into a routine of doing so and not go through the hassle of checking other sources of information, such as old paper maps. Quick decisions that require near real-time reaction time from the crew can easily be influenced by a false view of the surroundings, ultimately causing the crew to make the wrong decision. In the context of under keel clearance, the attacker would be interested in exploiting threat categories TC-1, TC-2, and TC-8. A sequence diagram showing a possible message exchange is shown in Figure 7.10

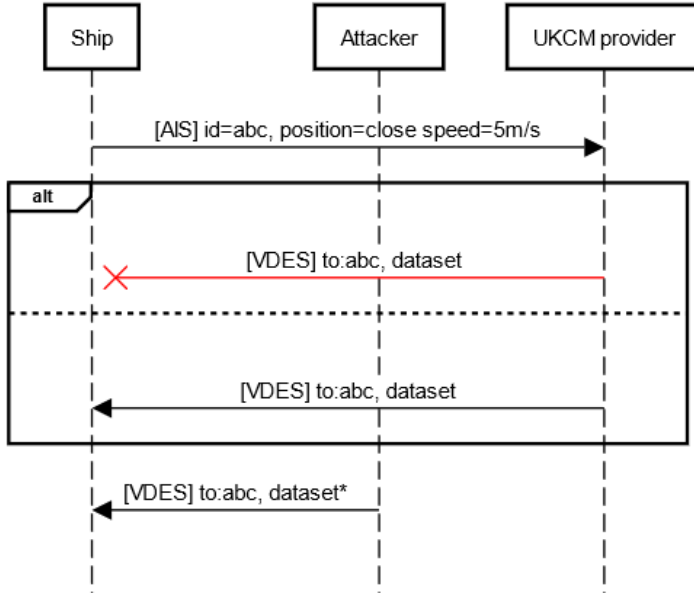


Figure 7.10: Data set forgery sequence diagram

Reconnaissance

First, the attacker must gather information about the victim ship to tie the attack to each ship. The MMSI is the ship's unique identification and associates the ship with a list of publicly available information such as the ship's position, heading, speed, weight, draught, estimated route, Estimated Time of Arrival (ETA) etc. This is valuable information for an adversary who wants to forge this information during the attack. Specifically, draught and the estimated route are essential in the case of UKCM, as this tells us the deepest point of the ship beneath the water surface and if the ship is headed toward dangerous waters. For instance, by researching the Marine Traffic website, the attacker can gather much information about potential victims. An example is shown in Figure 7.11, where ship draught, estimated route, ETA, and speed is available.

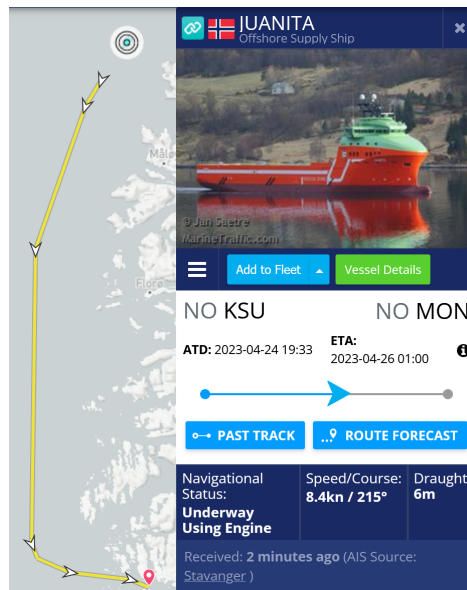


Figure 7.11: Screenshot from marinetraffic.com

The RCM for the reconnaissance stage is shown in figure Figure 7.12, which includes finding a suitable victim. Suitable victims could be ships with destinations inside waterways with a high risk of grounding, making any disturbance to the view more critical.

Marine Traffic offers premium solutions for a fee, where the most expensive plan gives advanced access to nautical charts, traffic congestion, density maps, and unlimited tracking of ships over satellite. The free plan would, however, be enough for gathering information for this attack. Other ways, such as bribing crew and phishing, could also be employed, but we assume this would be more costly.

Finally, the attacker could in theory also gather information via social engineering techniques, such as bribery and phishing. How much the bribe should be is difficult to estimate, however we would argue that a maximum of \$5000 comes a long way. Therefore we set a low confidence for the cost of this alternative. On the other hand, phishing could be seen as “free”, given that the attacker only uses free methods, such as email.



Figure 7.12: Reconnaissance stage for UKCM

Weaponisation

During the weaponization phase, the attacker must develop a malicious payload, which is delivered in the next stage. The malicious payload is a data set that shows a false view of the waters, potentially causing mariners to make wrong decisions. This can create dangerous situations, especially for mariners unfamiliar with the area. Figure 7.13 shows the RCM for this stage. The cost of intercepting the data depends on if the attacker decides to make a VDES receiver from scratch, which we can estimate to be around eight weeks of work, or buy a VDES module from a vendor, which cost around \$4000. We also estimate that creating a data set from scratch would take between one or two weeks. After reviewing the standards, we are fairly confident that this task is possible in this time frame.

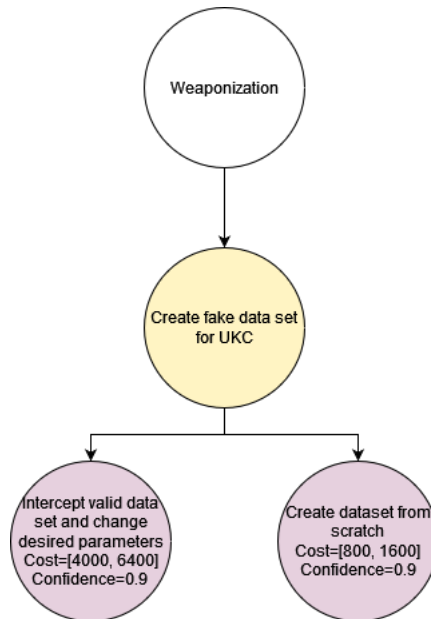


Figure 7.13: Weaponization stage for UKCM

Delivery

For the payload to have any effect, the payload has to be encapsulated in a message which is to be transmitted over VDES and finally received by the victim. The adversary has a few options for delivery. However, buying or stealing a VDES module could be the best option. Dumpster diving could yield old or obsolete technology and hard drives with data, including software that can facilitate the reverse engineering of proprietary solutions. In addition, Diversion theft is a social engineering method where attackers steal deliveries of packages, for instance, by attacking delivery services and modifying the destination coordinates [Hag20], which could be used against customers of VDES distributors. Figure 7.14 shows the delivery stage for UKCM. The payload can be delivered to the victim via satellite because of the introduction of VDE-SAT. [LHD22] showed that it is possible to send malicious commands to a satellite by the use of a Universal Software Radio Peripheral (USRP), which in the context of VDE enables an attacker to reach a larger number of ships, without having to be physically close to them. Permission to transmit over VDE is given to the transmitter by first sending a resource allocation request [IAL22]. The cost of this stage lies between 0 and 4000, depending on if the VDES module was acquired in the previous step or if the module has to be developed from scratch (most expensive). The most promising alternative is to buy a SDR (roughly \$360) and develop a VDES framebuilder, as done in our experiment (roughly \$800, given our assumptions).

We also wish to note that web-based SDRs exists⁶, where the user can listen to arbitrary frequency ranges free of charge⁷, but we did not explore this further in this analysis, as only receiving is currently available. However, one could imagine that transmitting capabilities would be available as a service in the future. This could benefit an attacker, as the web-SDRs are geographically spread out, potentially extending the reach of an attacker.

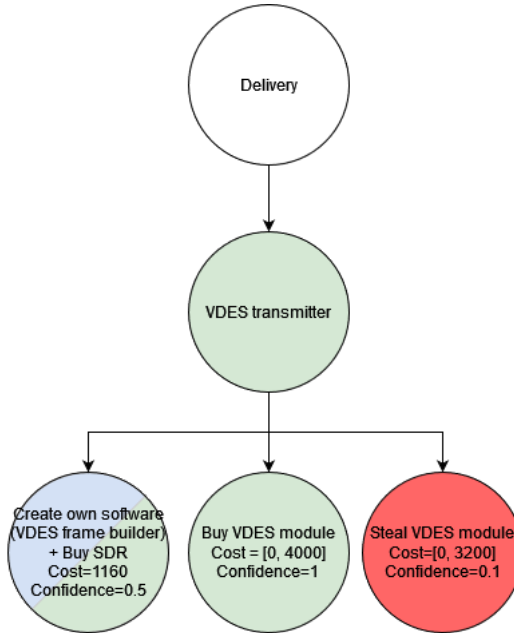


Figure 7.14: Delivery stage for UKCM

Exploitation

The receiving vessel investigates whether the chart is valid before accepting the payload. Today, charts are authenticated by ENC providers signing, using a 512-bit and 160-bit public/private key pair, and it is feasible to break cryptographically as discussed in Section 2.3. If we use Equation (2.1) for the 1024-bit key, we get an approximation of 4.4×10^{23} instructions, which is equivalent to 1.4×10^{16} MIPS-years. In this scenario, one AMD Threadripper must run for 6 billion years. Depending on which key is being used, the cost can be anywhere from 5000 (cost of AMD Threadripper) or infinitely high.

⁶<http://websdr.org/>

⁷<http://websdr.us:8902/>

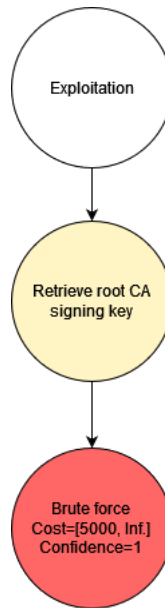


Figure 7.15: Exploitation stage for UKCM

Installation

The installation step of the UKCM attack relates to the victim parsing the data, which is subsequently shown to the crew via the ECDIS. This step is closely related to the weaponization step, as the data needs to be formatted correctly to be executed on the victim system. However, the ship expects the ENC's to be encrypted using 5-byte random cell keys, per the S-63 standard [IHO20]. The cell keys are found in a cell permit and retrieved from a data server by providing a user permit. The user permit consists of a 5-digit hexadecimal HW_ID, which is encrypted using the Blowfish algorithm, and a 5-digit hexadecimal M_KEY, in addition to a CRC checksum and Manufacturer ID (M_ID). In this scheme there are $16^5 = 2^{20} = 1048576$ possible HW_IDs, similarly for possible M_KEYS. To forge a user permit, the attacker would at most have to perform $2^{20} * 2^{20} = 2^{40} \approx 1 \times 10^{12}$ evaluations of the blowfish algorithm, which is less than 1 MIPS-years ($60 * 60 * 24 * 365 * 10^6 = 3.12 \times 10^{13}$). In comparison, the AMD Threadripper can perform 2.36×10^{12} instructions per second, so it would take in the order of seconds to forge a permit. The attacker could also try to brute force the Cell key, which also would require 2^{40} evaluations, but which would only give access to one cell, so forging the user permit would give more foothold, as it would give access to more cells.

In the new recommendations for the S-100 standard, these values are increased as discussed in Section 2.3, and we estimate the cost of forging them here. To forge the

permit, the attacker would need to use an exhaustive search on combinations of the HW_ID and the M_KEY. With 2^{128} possible HW_IDs, and a similar number of M_KEYS, the total number of combinations is $2^{256} \approx 1.16 \times 10^{77}$, which is infeasible to search exhaustively today.

The cost of this step also depends on what keys are being used. However, today's keys could be exhaustively searched on a conventional computer, so we set this cost to zero. However, when the new keys come into play, the cost jumps to infinity. The final RCM is shown in Figure 7.16.

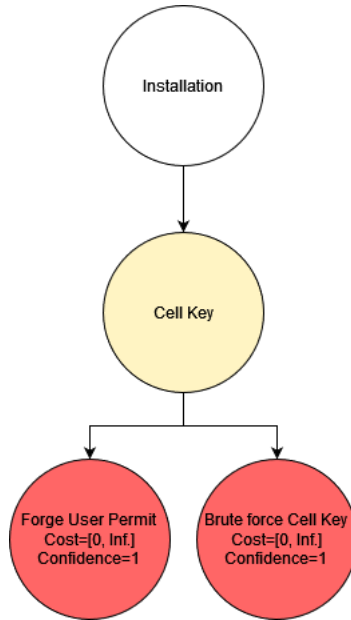


Figure 7.16: Installation stage for UKCM

Command and control

Forging a single dataset is insufficient to control the victim, as new updates can be issued every 60 to 10 minutes [IHO19], depending on the conditions. For the attack to have a persistent effect, the valid data sets from the service provider must be halted. If updates are continuously received, the ship could have a false view of the surrounding area of a maximum duration of 60 minutes, which depending on the circumstances, could be long enough to create a sufficient amount of confusion aboard the ship. Figure 7.17 shows the Command and Control stage RCM. To preserve the false view beyond the 60 minutes, the attacker must overpower the service provider by sabotaging or continuously overwriting the data shown on the ECDIS. Figure 7.10 also shows how jamming would be optional in this case. Compared to the other steps,

the command and control step is quite cheap, as it only entails repeatedly sending malicious data to the victim. The cost of buying a jamming device is assumed to be in the same price range as a VDES module.

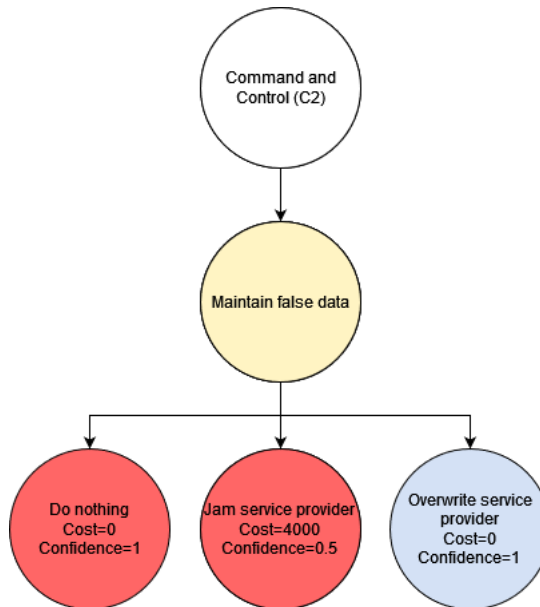


Figure 7.17: Command and control stage for UKCM

Actions on objectives

For the final stage, to fool the end users, the parameters should be chosen wisely not to raise any alarms. This involves changing the values within acceptable margins by considering other onboard systems such as AIS, GNSS, and wind measurements. The forged data set cannot show that the weather suddenly becomes worse if, for instance, wind measurements do not show the same change. Specialized software may therefore be needed to continuously calculate values to align with realistic observations made on deck. Figure 7.18 shows the RCM for this stage and points out that manual adjustment of these values is also possible. Still, the attacker would require domain-specific knowledge to say anything about how the parameters should evolve to avoid alerting the crew and assuming that it would take between 1 and 2 weeks to complete the specialized software and around one week to learn the thresholds of the vessel.

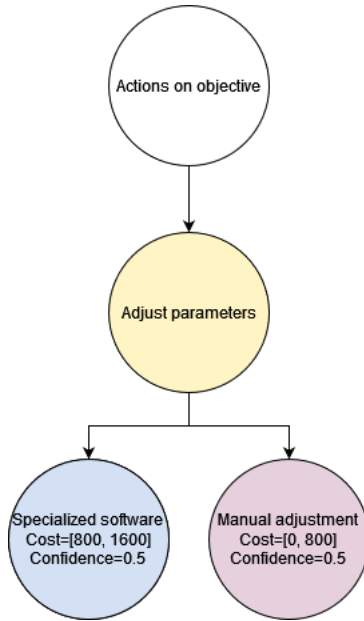


Figure 7.18: Actions on objective stage for UKCM

Cost estimation

Finally, the total cost of the data forgery attack is shown in Table 7.6.

Table 7.6: Total cost of Data forgery attack

Stage	Min Cost	Max Cost	Conf
Reconnaissance	0	5000	0.76
Weaponization	800	6400	0.9
Delivery	0	4000	0.53
Exploitation	5000	∞	1
Installation	0	∞	1
Command & Control	0	4000	0.83
Actions On Objective	800	1600	0.5
Total	6600	∞	0.15

7.2.5 Attack: Manipulate ETA

Description

This attack has been carried out in Section 7.1 and follows the same requirements and limitations. If the service provider demands an authenticated message, the requirements and constraints are the same as in Section 7.1.2. The attacker can decide to intercept the valid AIS message and change the information regarding the position and speed of the ship, causing the service provider to estimate a different ETA and consequently issue clearances for a different time window, for instance during high tides. The attacker could also change information related to the movement of the ship, for example, by spoofing that the ship is not moving, which can cause the UKCM provider never to issue detailed updates, as these should be provided when the ship is close to the service area [IHO19].

Both these attacks require the attacker to stop the valid AIS data from being received by the service provider and issue a different AIS message on behalf of the ship, causing the service provider to deliver the wrong UKCM data, or at the wrong time. The same analysis can therefore be used for both scenarios. The attacker would be interested in exploiting threat categories TC-3 and TC-7 for this attack. A sequence diagram showing a possible message exchange is demonstrated in Figure 7.19

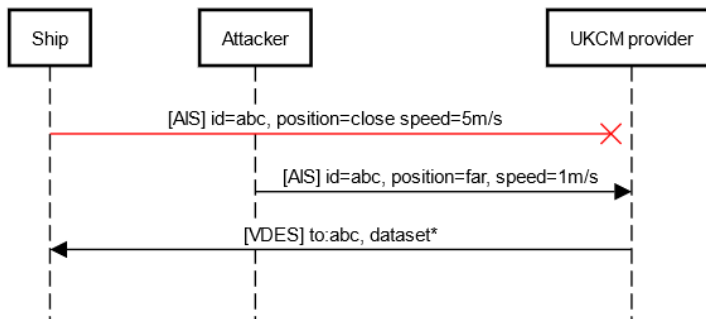


Figure 7.19: ETA manipulation attack sequence diagram

Reconnaissance

This step is very similar to the analysis done in Section 7.1.1, and the required resources for this step are shown in Figure 7.1. The only difference is that in addition to position, for UKCM, other values would need to be spoofed, such as speed and heading, but this is a trivial modification, so we consider the analysis done in Section 7.1.1 to be sufficient to show the execution of this stage. The cost for this stage is equivalent to the cost of the marine traffic subscription.

Weaponization

This stage is also similar to the weaponization stage of Section 7.1.1, so we refer back to this for this attack as well, and the RCM for this stage can be found in Figure 7.2. The only difference is that this stage requires spoofing of vessel movement, so our own AIS frame builder is more suitable for this application. Writing the frame builder took about a week, so 40 hours of work.

Delivery

This attack's delivery step is identical to the delivery step of Section 7.1.3, where the attacker can spoof from land via VDE-SAT retransmission. The RCM for this step can be seen in Figure 7.9.

Exploitation

If the AIS data is not authenticated, as in the AIS analysis of scenario 1, the service providers use false data to decide what to do next. However, if the service provider demands authentication, it becomes much more difficult, as in Section 7.1.2. Therefore, we set the minimum cost of this step equal to the cost of the exploitation step in Section 7.1.1, and the max cost to the exploitation step in Section 7.1.2. The RCM and for this step is also identical to Figure 7.4.

Installation

Malicious data needs only to be received and parsed by the service provider. Since AIS messages are parsed automatically, no further steps are required to facilitate the reception of messages. No actions are necessary, so this step's cost is also 0.

Command and Control

For the attack to persistently affect the victim ship and victim service provider, it is necessary to stop valid AIS data from the ship from ever reaching the service provider. Here the attacker can choose between jamming the AIS transmitter of the vessel or continuously overwriting the AIS messages. If we assume that a jammer is in the same price range as the VDES module and that the attacker can plant the jammer on the vessel by gaining physical access to the ship while it is stationary or by bribing with \$6000, the total cost comes out between \$0 and \$10000. The final RCM is shown in Figure 7.20.

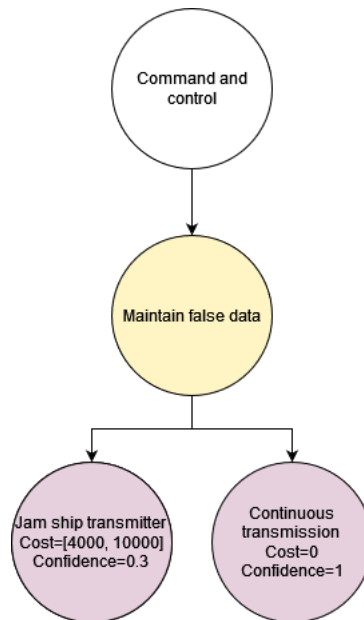


Figure 7.20: Command and Control step for ETA attack

Actions on Objectives

The continuous natural evolution of the false position, speed, and heading. This has shown to be possible in our experimentation with AIS, with the help of our frame builder. A specialized script needs to be created to simulate the natural environment of the ship and how it is possible to move. To write the specialized script anywhere from 1-2 weeks or 40-80 hours is a reasonable estimate. The total cost comes out between 0 and 1600 dollars. The RCM is shown in Figure 7.21.

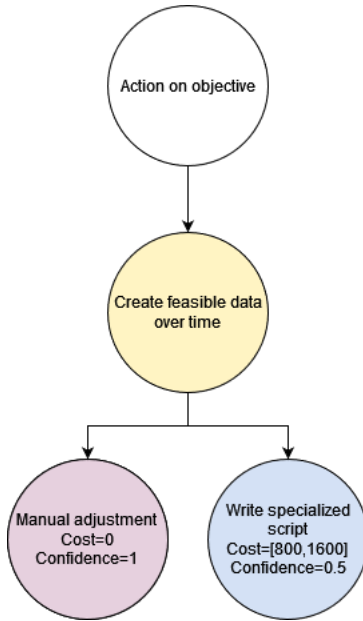


Figure 7.21: Actions on Objective step for ETA attack

Cost estimation

Finally, the total cost of the ETA attack is shown in Table 7.7.

Table 7.7: Total cost of ETA attack

Stage	Min Cost	Max Cost	Conf
Reconnaissance	0	312	1
Weaponization	800	800	1
Delivery	359	1600	1
Exploitation	0	∞	1
Installation	0	0	1
Command & Control	0	10000	0.65
Actions On Objective	0	1600	0.75
Total	459	∞	0.49

7.2.6 Attack: Update denial

Description

In this context, a denial-of-service attack aims to stop the information flow to the ship from the service provider. In the case of under keel clearance, updates to the pre-plan called the actual plan, which is more detailed, and possible updates to the actual plan, are intentionally blocked from reaching their final destination. An attacker could transmit radio signals at the same frequency as the VDE communication to cause interference. The attacker would be interested in exploiting the threat category TC-7 for this attack. A possible message exchange sequence diagram is demonstrated in Figure 7.22. To lower suspicion, the exchange also includes an optional AIS message to the service provider, consistent with the route plan.

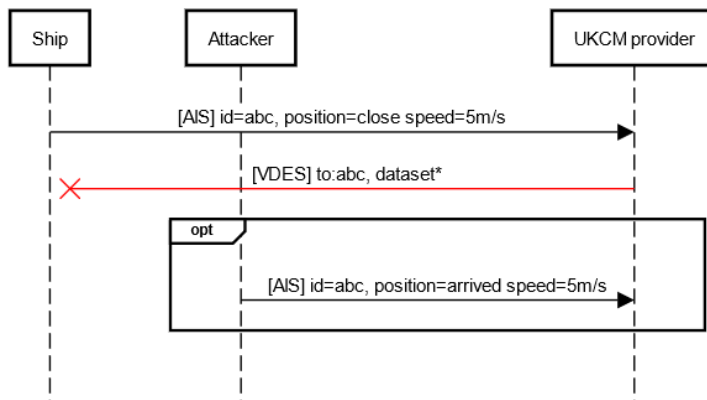


Figure 7.22: Update denial sequence diagram

Reconnaissance

This is similar to the step described in Section 7.2.4, where information about the ships' position, speed, location, etc., can be retrieved from online sources. To carry out a denial-of-service attack, figuring out what radio frequency the victim uses is also needed. The VDES maritime bulletin board is broadcast from VDES base stations and tells what frequencies are being used but not what ships are using them. Finally, jamming must be performed before updates are issued, so the estimated arrival point of the ship is also needed.

Weaponization

Weaponization relies on creating noise, so here, the payload can be random bytes. Since no specialized data is required, this stage's cost is zero.

Delivery

It must be physically close to the vessel to be within jamming range and stop valid AIS data from the service provider from reaching the ship. Here a jammer must be acquired and within range of the victim receiver. To accomplish this, we assume that a bribe of \$5000 could come a long way or that it would be possible to rent a boat for \$1000. Finally, physically sneaking onboard the ship while stationary could also be an option. Here we have estimated the cost to be equivalent to purchasing some disguise in addition to between 1 and 2 weeks of planning and finding out when and where the ship is stationary.

Regarding the hardware, we assume, as before, that a jammer can be purchased for a similar price to the VDES module. The RCM for this step is shown in figure Figure 7.23. The most significant cost of this step is acquiring the jammer and placing it close to the ship. We also assume that the jammer’s power needs to be estimated, and we have assumed that between 1 and 2 weeks of work is required.

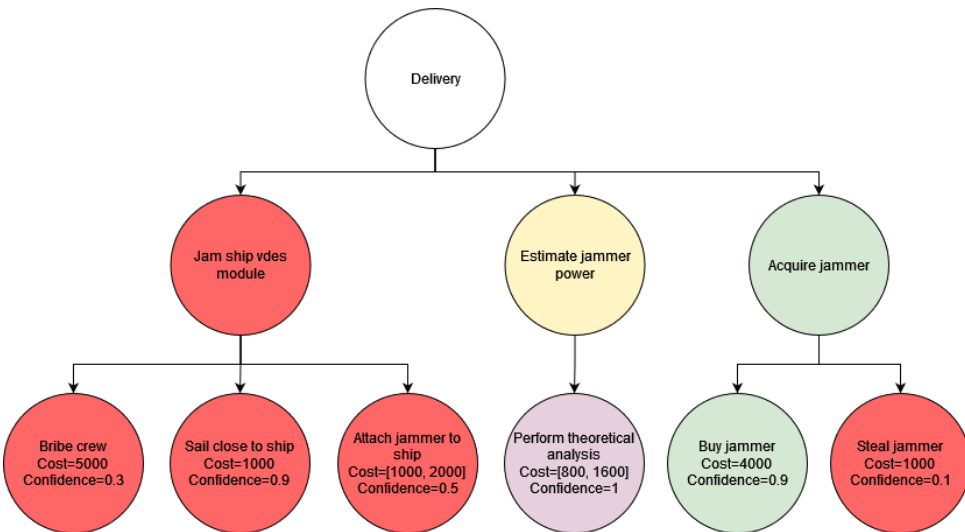


Figure 7.23: Delivery step for Update denial attack

Exploitation

No communication is longer possible as all TDMA slots are used, and there is no way to stop the attacker from broadcasting. The attack is triggered when the jammer is activated, so no cost is related to this step, and the RCM is irrelevant.

Installation

The receiver is not able to distinguish valid communication from the noise, which means that all received data is be garbage. To accomplish this, the jammer must be within reception distance of the victim. This is closely related to the delivery stage, and the cost for installation is covered in fig. 7.23.

Command and Control

The denial-of-service attack would be detectable, and methods of triangulating the signal are available to authorities. To deploy this attack effectively, the attacker would have to plan the attack carefully so that neither the vessel nor the service provider would have time to react and fall back to other solutions. To minimize the time the attacker is broadcasting, the attack would have to start right before the vessel receives the updates and go on for as long as the updates were provided. Timing this would be a challenge, and specialized software could be needed. We assume that such software could be developed between 1-2 weeks of work, giving the following RCM in Figure 7.24.

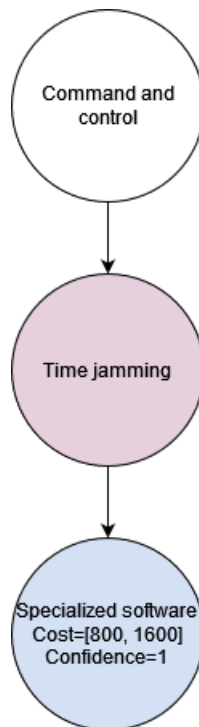


Figure 7.24: Command & control step for Update denial attack

Actions on Objectives

In the end, the attack would effectively stop the vessel from receiving updates in the time window where it is most critical. Consequently, this would increase the chance for the victim vessel to ground, as it would not be able to see that the environment has changed since the last update. This attack is most efficient when a significant difference exists between the outdated view of the clearance area and the most up-to-date data sets. A good example is when the ship believes the water currents are weak when they are strong due to recent shifts in the weather. Targeting areas where the weather is unstable would yield the highest variations between data sets. This leads to datasets becoming stale quickly. Publicly available weather data would be interesting here, so no further cost is associated with this step.

Cost estimation

Finally, the total cost of the update denial attack is shown in Table 7.8.

Table 7.8: Total cost of Update denial attack

Stage	Min Cost	Max Cost	Conf
Reconnaissance	0	312	1
Weaponization	0	0	1
Delivery	2800	10600	0.28
Exploitation	0	0	1
Installation	0	0	1
Command & Control	800	1600	0.9
Actions On Objective	0	0	1
Total	3600	12512	0.16

7.2.7 Attack: Exchange set replay

Description

Service providers broadcast valid exchange sets containing clearance data at different times of the journey and can be picked up by anyone. As discussed previously, the data sets are encrypted as the client needs to purchase certificates to decrypt them, ensuring that no charts can be exchanged between ships. Replaying this data would still mean that only one ship could decrypt it. In the case of such an attack, the ship could decrypt old information and potentially use this for planning. It would also cause the crew not to fall back to other systems for navigating the UKCM area. For this attack, the attacker would be interested in exploiting threat categories

TC-1, TC-2, and TC-7. A sequence diagram showing a possible message exchange is demonstrated in Figure 7.25. In the exchange, the attacker blocks the new data set *dataset**.

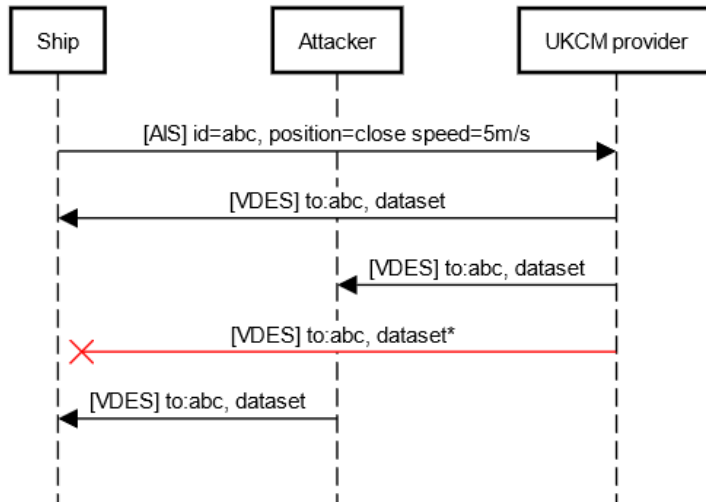


Figure 7.25: Replay attack sequence diagram

Reconnaissance

During the Reconnaissance phase, finding out when the vessel is arriving at the designated clearance area is of interest to the attacker. Therefore, this phase's cost estimation would be identical to the one found in Section 7.2.4.

Weaponization

For this step, intercepting a valid data set is required. This can be done with a VDES receiver, and the cost associated with this step is identical to the RCM found in Section 7.2.4.

Delivery

The attacker needs to deliver the valid data to the same ship at a later time and stop service provider data from being received. The process of stopping valid data sets has been analyzed in Section 7.2.6, and we refer to the RCM conducted there for the cost estimation.

Exploitation

According to [IHO19], one of the attributes of the UnderKeelClearancePlan is *fixedTimeRange*, which is a period for which the data set is valid. It is uncertain what

values are intended for this attribute, as it might depend on how fast the weather shifts. For stable weather, a long time range could be set. However, revoking this validity after it has been issued is impossible. However, this should not be mistaken as a countermeasure against replay attacks since an attacker could still replay data sets within the time window. Furthermore, it is uncertain what happens if a data set is received outside of the given time range, and we can assume that this is left to the application layer logic. Therefore, no additional actions are needed for this step.

Installation

For the data to be interpreted by the receiver, the victim needs to be able to decrypt it. However, this is not an issue since the data set is already encrypted with a valid key. The victim has no choice but to decrypt and parse the valid data set. This step does not require any further involvement from the attacker, and the cost for this step is 0.

Command and Control

To have a persistent effect on the victim, the attacker also needs to prevent valid data from the service provider from being received by the victim, as this would cause updates to be overwritten. This stage, therefore, is identical to the Command and Control stage of Section 7.2.4, so we refer to this section for further cost analysis.

Actions on Objectives

When the vessel moves closer to the service area, the crew has expectations for when the first data set arrives and the frequency of future updates. It might be necessary for the attacker to continue replaying the data set. It is worth mentioning that this could raise suspicion, as the victim vessel would consequently receive the same data set multiple times. However, it is uncertain what checks and alarms application developers implement. Ultimately there is not much the attacker can do at this point, so no additional cost is added to this step.

Cost estimation

Finally, the total cost of the data replay attack is shown in Table 7.9.

Table 7.9: Total cost of Data replay attack

Stage	Min Cost	Max Cost	Conf
Reconnaissance	0	4000	1

Continued on next page

Table 7.9: Total cost of Data replay attack (Continued)

Stage	Min Cost	Max Cost	Conf
Weaponization	800	6400	0.9
Delivery	2800	10600	0.28
Exploitation	0	0	1
Installation	0	0	1
Command & Control	0	4000	0.83
Actions On Objective	0	0	1
Total	3600	25000	0.21

7.3 Results from AIS and VDES analysis

Table 7.10: Estimated total cost results

Scenario	Min Cost	Max Cost	Conf
Scenario 1: Optional authentication	538	18520	0.418
Scenario 2: Mandatory authentication	∞	∞	0.22
Scenario 3: Retransmit over VDE-SAT	538	13520	0.76
UKCM data forgery	6600	∞	0.15
UKCM ETA manipulation	459	∞	0.49
UKCM update denial attack	3600	12512	0.16
UKCM replay attack	3600	25000	0.21
Scenario 1: Walde and Hanus	1840.8	13690.8	0.1

Scenario 1: Optional authentication

The assumptions in Scenario 1 and the assumptions used by Walde and Hanus are similar. From the cost table table 7.10, we can see that our min cost and max cost are lower and higher, respectively. The minimum cost is lower due to our custom AIS transmitter, which we consider easier to use than AIS_TX. We have also excluded acquiring a computer since we assumed anyone trying to spoof AIS messages would already have one. If we were to include it, using the min cost value from Walde and Hanus \$300, the minimum cost would rise to 838, which is still cheaper than the total cost they calculated. Interestingly enough, the maximum cost is also increased. This is because, due to our AIS experiment, we could set a cost associated with building and writing our own software. If the assumptions set under Scenario One are used in a real-world setting, the result of our AIS experiment would therefore lead to a decreased cost associated with spoofing AIS messages.

Scenario 2: Mandatory authentication

As explained in section 7.1.2, VDES has identified an example protocol to transfer the signature of AIS messages over VDE-TER. This is shown in Scenario Two, where the cost of conducting AIS spoofing attacks is set to infeasible. This is because the cryptographic scheme used to generate a signature in VDES is cryptographically secure. In reality, the minimum cost would not be infeasible, as there are different ways of obtaining a private key, as explained in section 7.1. In this scenario, an adversary would also need a VDE-TER transmitter that can transmit the signature of the AIS message. Depending on the communication state, the adversary might

also need a VDE-TER receiver to receive the resource allocation response from the target.

Currently, no open-source software is available online that enables such an attack to occur. Despite this, we believe the first piece of work has been laid down by creating the first open-source VDES frame builder. By having a possible signal simulator ready, it is also easier to imagine how the receiver must look to undo the processing done on the data before transmitting. Therefore, in scenario two, the cost of spoofing AIS messages increase because of the authentication scheme and the increase in complexity due to VDE-TER.

However, sending two related messages over separate communication channels can be bad for several reasons. If an adversary acts as a Man-in-The-Middle (MiTM) and modifies the content of the AIS message, the receiver can understand that an attack is occurring by checking the validity of AIS message. In this case, the authentication scheme works as intended, and the recipient can disregard the modified AIS message.

If the adversary instead modifies the signature, the recipient might disregard a valid AIS message. This attack differs from traditional Distributed Denial of Service (DDoS) attacks, as seemingly the security scheme works as intended. Therefore, such attacks are harder to detect, and two ships might disregard the AIS messages and head toward a collision. As discovered by Walde and Hanus, such attacks would likely not cause collisions in the case of manned ships, as ships have different ways of communicating with each other [WH20]. In the case of autonomous ships, the result might differ, and a potential accident might occur if the ship disregards valid AIS messages.

TESLA as an authentication mechanism

TESLA was mentioned as a possible authentication protocol. As discussed in Section 2.5.3, TESLA only proves a message originates from the same source, not the identity of the original sender. Therefore, a PKI must be available for a vessel to validate its identity. The main benefit of the TESLA protocol is that the overhead is significantly smaller, as not every AIS message needs to be authenticated with its own signature. Therefore, the scheme's security would rely on the PKI, and the cost would relate to Scenario Two.

If all AIS messages were to be authenticated, an attacker could send out fraudulent signatures at a large rate. Because of the computational cost of verifying digital signatures, this could lead to a form of DDoS attack, as the recipient would be busy verifying fraudulent signatures [PCTS02]. Since the security of the TESLA protocol relies on the security of the PKI, the cost would remain the same as in Scenario 2.

Scenario 3: Retransmit over VDE-SAT

An example protocol format for retransmitting AIS messages over VDE-SAT to augment the communication between ships and between ships and shore has been identified. From the results visualized in Table 7.10, we can see that the minimum cost remains the same as in Scenario One, but the maximum cost decreases. If an AIS message is forwarded from shore via VDE-SAT, the attacker would no longer need to travel in a boat to be within the transmission range of the target. Even though this feature is meant to augment maritime communication, it might make it cheaper for an attacker to spoof AIS messages by reducing the adversaries' cost while increasing the range of the attack. Enabling VDE-SAT to retransmit AIS messages might therefore increase the number of AIS spoofing attacks because the cost associated with the attacks is decreased.

UKCM data forgery

Table 7.10 shows that this attack may be impossible to carry out, given that the new S-100 data protection scheme is deployed. For this attack, the cyber kill chain is broken in the exploitation step, as it would be infeasible to break the 1024-bit DSA key with today's methods. The attack requires the use of the old S-63 protection scheme to be feasible, which remains in operation in the years to come. This is captured by the minimum cost, which is still high but no longer infeasible. Data sets in the old scheme are consequently possible to forge for attackers with moderate to high budgets. This would typically be hacker groups or even state actors. It is also important to mention that even though the S-100 protection scheme is infeasible to break today, it becomes outdated again beyond 2031, as discussed in Section 2.3 [Bar20]. Bearing in mind that updating these standards may be difficult and time-consuming, S-100 data sets might be vulnerable for some time in the future.

UKCM eta manipulation

This attack is one of the more feasible attacks, with the lowest minimum cost amongst the UKCM attacks presented in Table 7.10. The main vulnerability in this service is the use of AIS for keel clearance calculations, which we have seen cannot be trusted without additional protection. An additional cost of this attack is related to how an attacker would prevent valid AIS data from reaching the service provider, raising the maximum cost significantly. In our analysis, a couple of attack vectors were explored. However, more elaborate attacks could be even more efficient, for instance, by first carrying out a frequency hopping attack as discussed in Section 2.1.2, disabling the AIS transmitter.

UKCM update denial

From the results, this attack is also shown to be feasible under the right conditions. Table 7.10 shows that there is some uncertainty related to the costs, as the attack relies on disrupting the VDES module of the victim ship, and we have not yet seen this done in practice. The main challenge for the attacker in this scenario is to be within reception distance of the ship and also start the jammer at the correct time before more detailed updates to the UKCM plan are issued. Hence, the cost is not necessarily very high. However, the rate of success varies based on external factors such as timing. To increase the rate of success, the attacker would probably increase the number of victims, for instance, by jamming maritime frequencies in a high-traffic area.

UKCM replay attack

The rate of success for this attack depends on the detection mechanisms present in the application layer for detecting duplicate UKCM plans. The crew also needs to decide what to do if an outdated plan is received when they have no other alternative plan. External factors such as timing have an impact on the success rate, and since a given UKCM plan can only be read by one single ship, an attacker would also be required to collect multiple plans for different ships to scale the attack. The attacker is also dependent on dangerous environmental changes, which is information that the attacker wants to withhold from the crew. Therefore, it is unlikely that such an attack could be carried out, even with an unlimited budget.

7.4 Challenges of the RCM

The RCM is a valuable methodology for coming up with a rough estimate of the feasibility of attacks. The main benefit of this methodology is that it aids the analyst in reasoning about the system in question and pinpointing possible areas of vulnerability, which is further explored using other methodologies, such as STRIDE. However, it still has some significant drawbacks that only make it suitable for discussion, not as an exact science.

The first major challenge with this method is that cost and confidence estimates are highly variable when different people do the analysis. The best example is when the attacker has to develop code to complete some kill chain steps, which takes some unknown time. Depending on the attacker's skill level, this can vary significantly from a few days to months. Also, the cost per hour of development is highly uncertain, as it depends on if the attacker has other financial opportunities available or the attackers' Opportunity Cost of Capital (OCC). For the software to represent some cost, realistically, the attacker would have to give up some other alternative that would pay off in the form of capital gain. The cost of developing the software is the

equivalent of the loss of this opportunity to create capital. However, highly motivated lone hackers might not have anything better to do after work and could use their spare time to develop this software. Approximating how difficult it is to develop specialized software depends on many factors, and different people can come up with different results.

The second challenge with the RCM is that it does not consider that equipment can be reused over multiple steps. According to the methodology, the total cost of the attack is based on all the cheapest alternatives for each step and all the most expensive for each step. This could yield a sub-optimal solution, as resources can, in reality, be reused across multiple steps in the kill chain and multiple attacks. It would therefore be misleading to say that the total cost of conducting multiple attacks is simply the sum of the individual attacks. The best example is in the form of hardware that is used to transmit or receive signals, which is only required to be purchased once but could be used in all attacks dependent on these resources.

A third challenge when using the RCM is that the RCM does not consider the scope of the attack. An example is when the attacker attempts to spoof the vessel's movement. Values must be within certain thresholds based on the ship's capabilities to avoid raising alarms. The cheapest approach is to not care about these thresholds, which might make the attack less effective. In some cases, the attacker is also required to purchase some hardware. This could leave a paper trail, making it possible for organizations investigating the attack to pursue the attacker. These hidden costs may arrive after the attack, affecting the attacker's well-being, which they may or may not be ready to sacrifice.

The resource modeling for each step can also present some challenges because each step in the kill chain is often more complex than using simple AND and OR operations. For instance, doing nothing is sometimes a valid action for a step, even though it might yield worse results. This is seen in the command and control step example when forging UKCM data sets, where an optimal solution is maintaining control over the victim by continuously overwriting valid data sets from the service provider. However, the attacker can skip this step and only send one forged data set, which could be overwritten by a valid data set at any time. This ultimately depends on the scope of the attack. If two resources were needed for this step, with an AND clause between them, choosing the "do nothing" alternative effectively disables this resource, consequently turning the AND clause between resources into an OR clause. This breaks the rule of the RCM that states that resources must have AND clauses between them.

Finally, resource classes such as "Skill", "Tangible", "Logic" etc. bring little information to the table and are often more complex than one single class, making

the class itself somewhat arbitrary. For instance, if the attacker requires a transmitter to deliver a malicious payload, transmitters can be stolen, bought, or created using an SDR. Creating a custom SDR is both something that requires hardware and software but also something that requires skill and domain-specific knowledge. It can therefore be misleading to choose one class (e.g., “Tangible”), as this can indirectly signify that another class is not needed (e.g. “Skill”).

Despite all these challenges, the RCM can still be a helpful tool that can be used as a basis for a discussion on cyber security in the given system. Additionally, the interest of the RCM does not lie in the exact numbers but in orders of magnitude. Ultimately, finding out if an attack is possible, even if only feasible with an extraordinarily high budget, is information of high value to stakeholders and end users.

Chapter 8

Discussion

This chapter discusses the various results from our literature review, interviews, experiments and analysis. First, we put these results into the context of our research questions and give a definitive answer to each one. Then we discuss the ethics of our findings before we present the future work of this thesis. The future work section describes related research areas of interest that are out of the scope of this thesis but could be the topic of another.

8.1 Research questions

8.1.1 Q1: How will VDES change the security of maritime shipping industry compared with AIS?

Our analysis also shows that higher availability brings benefits in terms of connectivity, service coverage, and challenges regarding a broader attack surface. The current specifications as they stand today allow for unauthenticated retransmission of AIS messages over VDE-SAT, which lowers the cost of delivering malicious payloads. This consequently increases the probability of an attack being successful.

At the same time, some services such as UKCM and other S-100 data products still depend on AIS, which can give end users a false sense of security. Fallback procedures toward more analog technology are essential for the entire security picture. A healthy adoption of this new technology ultimately depends on the level of trust the end users provide. When time-critical choices need to be made, seconds lost due to confusion created by attacks against service providers could be enough to push the victims into dangerous situations.

VDE messages also differ from AIS messages as they can transfer arbitrary data, not only predefined data. This mechanism ensures that the security of the VDE messages is defined by the application security and not the security of VDE.

Therefore, to evaluate the security of VDE messages, the security of MCP and the S-100 standard has to be evaluated.

Another key aspect of trusted services is the difference between manned and unmanned vessels, where a ship with a crew can weigh risk differently based on contextual factors such as experience. This allows more nuanced conclusions to be drawn and to be skeptical even when no alarms have been triggered. A crew is also able to accept risk when benefits weigh heavier. In the worst-case scenario, a mariner can disregard all electronic systems and rely on manual navigation as seafarers have done for millennia. However, this is often done at the cost of speed and may not always be possible in time-critical situations.

The security of proprietary implementations is also difficult to verify, and possible attacks against VDES can therefore be difficult to reveal. As we experienced throughout our experiments, research also becomes more difficult when the necessary tools are missing to carry out a simulated attack. The work of creating open-source versions of proprietary implementations can boost the research, which benefits the industry as a whole.

Closing remarks on Q1

the complexity of VDES and MCP is higher than before, which makes management, monitoring, and maintainability more of a challenge, where more mistakes are possible. At the same time, VDES is a much-needed upgrade in comparison to AIS and does so by facilitating the implementation of new technology such as MCP, thereby increasing availability, integrity, and reliability. These are security mechanisms that are lacking in the current AIS implementation. Hence, we can confidently say that VDES improves the overall security in the maritime industry.

8.1.2 Q2: How will VDES impact the difficulty associated with attacking maritime communication services?

Regarding the transmission of AIS messages as discussed in Section 7.3, authentication increases the cost for an attacker trying to spoof AIS messages. As shown in Table 7.1, the proposed authentication scheme would make it infeasible to spoof AIS, as the security scheme used to generate a signature is sufficient. While using two channels enables transmission of AIS signatures, it is still possible to jam the integrity message and prevent it from reaching its desired destination.

As discovered in Chapter 4, authentication of AIS messages will most likely remain optional. The retransmission of AIS messages over VDE-SAT would therefore lead to a decrease in the difficulty regarding spoofing AIS as shown in Section 7.3.

This is because the cost of the weaponization stage is decreased due to our AIS, frame builder implemented in Chapter 5 and reduced cost regarding delivery.

Initially, the cost of the weaponization stage for VDE messages would be high, as no open-source software is available as of June 2023. However, as we have proven in Chapter 6, it is possible for adversaries to make VDE frame builders. Being a new technology, the initial cost for the first attackers would therefore be high, but as time goes on and more open-source material is available. The weaponization stage cost decreases as a consequence and, therefore, also the overall difficulty.

Regarding the exchange of Hydrographic data, the security scheme of the S-100-based products is sufficient, as shown in Section 7.2.4, and makes it infeasible to manipulate S-100-based data sets such as UKCM at this time. Regarding the attacks on UKCM, as explained in Section 7.2.4, it is clear such attacks involve complex steps that, even though they can be performed, introduce significant costs for the adversary. The highest cost is associated with the cryptographic keys, ensuring the data is encrypted and authenticated. Even though it is theoretically possible to break the DSA signatures, time and domain-specific knowledge in relation to the implementation of tools and software is still needed to carry out the attack.

It is also important to remember that it will take time for older systems to upgrade to VDES, which means they do not receive the cryptographic advantages of S-100 discussed in Section 2.3 right away. Legacy systems will continue to be vulnerable until they make the transition to VDES. It is still unclear to what degree early adaptors of VDES can download and use S-100-based data products, as S-100 and VDES are developed independently of each other. This means that users could potentially be forced to use S-57 compliant ENCs over VDES for a period of time. If this scenario is realized, end users would be even more exposed due to the reach of VDE-SAT.

As discussed in Section 7.2.5, UKCM uses AIS messages to calculate the ETA. AIS based attacks could therefore be applied to attack what is perceived as secure S-100-based data sets transmitted over VDES. This would ultimately make it easier for attackers with harmful intentions.

The results from the RCMs in Chapter 7 shows that the delivery stage generally has become easier to perform with the introduction of VDES. This increases the risk of successful attacks, such as AIS ship spoofing and the UKCM Denial-of-Service (DoS) attack, as they mainly rely on successful delivery. Additionally, when services such as UKCM rely on AIS, AIS-based attacks can be applied to VDES-based products. On the other hand, data forging is becoming more difficult because of stronger cryptographic mechanisms introduced by S-100 and MCP, decreasing the risk of spoofing in VDES-based services.

Closing remarks on Q2

VDES on its own does not paint the entire picture of the security landscape in the maritime industry. With the development of the MCP and services such as S-100-based data products, members and services can be mutually authenticated. One could also argue that the MCP could not be realized without VDES as an underlying delivery mechanism. Therefore, we can confidently say that the introduction of VDES ultimately strengthens the maritime industry's overall security and makes it more difficult to attack maritime communication services.

8.1.3 Q3: Will introducing VDES introduce new cyber threats in the maritime industry?

Our goal with this question was to see if any new threats have been introduced to the maritime industry. The result from our methodologies suggests that new threats from other domains are, in fact, simultaneously introduced with the introduction of VDES.

The results from the interviews and literature review show how VDES allows for increased automation, interconnectivity, coverage, and availability to both stakeholders at sea and on land. The capabilities of VDES stretch beyond only improving coverage and bandwidth but also manage to facilitate an entirely new ecosystem where maritime services work together to provide necessary data to workers at sea in time-critical situations. Additionally, VDES manages to bridge the technological gap between sea and land and enables the development of new and modern applications by introducing well-known concepts such as IP/TCP stack to vessels globally.

With the introduction of services provided over IP onboard ships, malicious actors with long experience in attacking web-based services could also start to target maritime services and, thus, the vessels utilizing them. Hence, the most common security vulnerabilities found on the web may start to migrate over into the maritime domain. Consequently, this provides an attacker with more opportunities, as potential victims at sea are reachable regardless of the victim's geographic location.

From our experiments on AIS and VDES, acquiring the necessary commercial off-the-shelf (COTS) and open-source software is today a feasible way of disrupting maritime services. This enables attacks from individuals and smaller groups with lower budgets. More available tools and cheaper hardware subsequently increase the capabilities of the attacker. Additionally, it is challenging to stop attackers from using tools for malicious purposes. Ultimately, we can anticipate these attacks to continue or become more prevalent unless resource-efficient authentication, such as the proposed TESLA protocol, is implemented and becomes mandatory.

We would also argue that an increasing intent to do harm is present due to the growing size of the maritime industry. An attack that incapacitates a large shipping vessel or service provider could have great economic consequences for shipping companies, providers, authorities, or local populations. Hence, both attackers with economic, political, or ideological motivations would consider these stakeholders attractive targets.

Closing remarks on Q3

Based on the above factors, if we consider an attacker or event to be a threat if intent, capability, and opportunity arise, we can confidently say that VDES attracts new types of threats. However, it is important to note that this does not necessarily increase the cyber risk, as this is also dependent on what vulnerabilities are identified and mitigated.

8.2 Ethics

The thesis contains explicit guides on spoofing AIS messages and a new embedded Python AIS frame builder block. As a result, the cost in terms of dollars decreases when spoofing AIS messages. The thesis also contains a guide on how to build a generic VDE frame builder according to the technical specification of ITU. Before this thesis, no open-source material regarding VDE was available online. The VDE-TER frame builder and possible signal simulator might also help possible adversaries spoof VDE-TER messages. Section 7.2.4, also contains a thorough analysis of the vulnerabilities in the old S-63 standard and the possibilities of the S-100 standard. Despite this, we believe that contributing with the frame builders according to the technical specification might help fuel new research in the context of VDES and enable future research to build on the material discussed in this thesis.

8.3 Future Work

As VDES is still in the testing phase, we have identified three possible scenarios regarding AIS messages that might be applicable in the future. After the introduction of VDES, new research should verify or dismantle the theories in this thesis.

The thesis contributes with VDE-TER frame builder and signal simulator. However, as no valid VDE-TER messages were available to test the frame builder on, we built a generic frame builder as explained in Annex 2 in [ITU22] and verified it for ASM messages. As explained in Section 6.1.9, the frame builder might contain flaws we are not aware of, and then, especially with regards to the endianness. When examples of VDE-TER messages are available in the future, research should be

conducted to verify if the frame builder works as intended and if it is possible to spoof VDE-TER messages using the setup explained in Chapter 6.

Future research should also be conducted to try and build a VDE-SAT frame builder. As a lot of the same methods used in VDE-TER are reused, we believe it should be possible to extend the VDE-TER frame builder to include link ID for VDE-SAT.

Different topics have been discussed throughout the thesis regarding the security of the MCP. Especially regarding the security of the MIR. What stops an attacker from simply breaking into a vessel and stealing the entity holding the private key associated with the ship? With the entity, an attacker can spoof AIS, messages which could be extended to ensure the target receives wrong and possibly harmful information from service providers, as discussed in Section 7.2.4. Future research is therefore needed to ensure that simply breaking into a boat and stealing the entity holding the private key is sufficient to spoof AIS signatures.

The vetting procedure has occasionally been mentioned throughout the thesis. The vetting procedure is currently conducted by the MCP instance. In the future, however, this might change so that each flag state conducts its own vetting. Therefore, future research should also be done on whether enlisting a malicious organization to different MCP instances is feasible.

Research regarding other S-100 standards and their use of AIS data should also be concluded. A cryptographically secure standard that relies on an unsecured data supply is insufficient. Research should be conducted to ensure future service providers providing security-critical S-100 products use authenticated AIS messages.

The impact of the different attacks might also differentiate between manned ships and autonomous ships. Future research is therefore needed to ensure that autonomous ships are not impacted severely by attacks discussed in this thesis. Attacks that would not necessarily impact ships greatly could be severe in the case of autonomous ships.

Finally, our improved AIS frame builder is well-suited for further customization, which makes more complex attacks possible. A few of these attacks have been used as a basis for our analysis. However, a practical implementation could yield interesting results. By designing an attack that also takes the physical characteristics of the vessel into account, the attacker could potentially circumvent some of the more basic Intrusion Detection Systems (IDSs), as suggested by [AOGK22].

Chapter 9

Conclusion

The goal of this thesis was to investigate the security of VDES and future VDES based systems and to see what lessons have been learned from the more mature and overloaded AIS system. The hope of VDES is to increase the safety, security, reliability, and availability of maritime communications by presenting a flexible and agnostic communication infrastructure at sea. Our research suggests that VDES does not directly address the security concerns prevalent in AIS. However, it becomes a potent instrument when it comes to navigational aid when paired together with other products such as the MCP and the S-100 family.

In pursuit of answering our research questions, we managed to identify a couple of key security concerns. Firstly, the only security upgrade to AIS is optional authenticated messages carried over a separate VDE channel. This concern is further substantiated by how future VDES-based services, such as the UKCM, depend on accurate and authentic AIS data to deliver dependable service. Secondly, the state of security in legacy systems was found to be poor compared to modern standards, and our analysis shows how forgery of signatures is possible under the right conditions. These legacy products continue to exist in the maritime domain and could potentially be misused on a larger scale as advancements in computing and cryptanalysis continue.

Our analysis and experiments show that the attack surface of maritime services expands as the complexity grows. Advancements in key areas such as computing, commercially available technology, and research in the field of cryptography are continuously providing attackers with new tools and methods. The maritime domain is entering a new environment with more interconnected systems and a higher degree of automation. This new environment exposes the industry to an entirely new class of attackers, able to conduct more complex types of attacks, such as the ones demonstrated in this thesis. For all future adopters, VDES ultimately represents a paradigm shift within modern maritime communications technology.

References

- [Alé21] Alén Space. «How will VDES services change ship tracking communications?» (Jun. 2021), [Online]. Available: <https://info.alen.space/how-will-vdes-service-s-change-ship-tracking-communications> (last visited: May 30, 2023).
- [AOGK22] A. Amro, A. Oruc, *et al.*, «Navigation data anomaly analysis and detection», *Information*, vol. 13, no. 3, p. 104, Mar. 2022, Number: 3 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: <https://www.mdpi.com/2078-2489/13/3/104> (last visited: Jun. 18, 2023).
- [ATSP20] A. Aziz, P. Tedeschi, *et al.*, «SecureAIS - securing pairwise vessels communications», in *2020 IEEE Conference on Communications and Network Security (CNS)*, Jun. 2020, pp. 1–9.
- [Bar16] E. Barker, «Recommendation for key management part 1: General», National Institute of Standards and Technology, NIST SP 800-57pt1r4, Jan. 2016, NIST SP 800-57pt1r4. [Online]. Available: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf> (last visited: Apr. 25, 2023).
- [Bar20] E. Barker, «Recommendation for key management: Part 1 – general», National Institute of Standards and Technology, NIST Special Publication (SP) 800-57 Part 1 Rev. 5, May 4, 2020. [Online]. Available: <https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-5/final> (last visited: May 2, 2023).
- [BBBM21] G. Bour, K. Bernsmed, *et al.*, «On the certificate revocation problem in the maritime sector», in *Secure IT Systems*, M. Asplund and S. Nadjm-Tehrani, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2021, pp. 142–157.
- [Bis21] Bistromath. «Create README.md · bistromath/gr-ais@2beab39», GitHub. (Mar. 2021), [Online]. Available: <https://github.com/bistromath/gr-ais/commit/2beab39aaddf731c215deb8354e8b1c0fa319058> (last visited: Jun. 5, 2023).
- [Bob15] S. Bober, «Paper 42 - AIS next generation – the development of the VHF data exchange system (VDES) for maritime and inland navigation», 2015.
- [BPW14] M. Balduzzi, A. Pasta, and K. Wilhoit, «A security evaluation of AIS automated identification system», in *Proceedings of the 30th Annual Computer Security Applications Conference*, New Orleans Louisiana USA: ACM, Dec. 8, 2014, pp. 436–445. [Online]. Available: <https://dl.acm.org/doi/10.1145/2664243.2664257> (last visited: Jan. 25, 2023).

- [Bra] E. R. Brand a National Instruments. «USRP b200mini», Ettus Research. (), [Online]. Available: <https://www.ettus.com/all-products/usrp-b200mini/> (last visited: Jun. 5, 2023).
- [Chi20] M. Chiappetta. «AMD threadripper 3990x review: A 64-core multithreaded beast unleashed - page 3», HotHardware. (Feb. 7, 2020), [Online]. Available: <https://hothardware.com/reviews/amd-ryzen-threadripper-3990x-cpu-review> (last visited: May 2, 2023).
- [CMRR23] L. Chen, D. Moody, *et al.*, *Digital signature standard (DSS)*, Feb. 2, 2023. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=935202.
- [Deb] Debian. «Socat(1) — socat — debian bullseye — debian manpages». (), [Online]. Available: <https://manpages.debian.org/bullseye/socat/socat.1.en.html> (last visited: Jun. 5, 2023).
- [EBHS16] T. Eriksen, L. E. Bråten, *et al.*, «VDE-SAT—a new maritime communications system», in *Proceedings of the Small Satellites, System & Services Symposium (4S), Malta*, vol. 30, 2016, pp. 1–12.
- [ESA23] ESA. «Satellite – automatic identification system (SAT-AIS) overview». (2023), [Online]. Available: <https://artes.esa.int/satellite-%E2%80%93-automatic-identification-system-satais-overview> (last visited: Jun. 5, 2023).
- [EUS16] EUSPA. «Expanding opportunities for maritime use of GNSS». (Jan. 15, 2016), [Online]. Available: <https://www.euspa.europa.eu/expanding-opportunities-maritime-use-gnss> (last visited: May 30, 2023).
- [FGR13] J.-C. Faugère, C. Goyet, and G. Renault, «Attacking (EC)DSA given only an implicit hint», in *Selected Areas in Cryptography*, L. R. Knudsen and H. Wu, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2013, pp. 252–274.
- [For22] J. Forsberg, *Cybersecurity of Maritime Communication Systems : Spoofing attacks against AIS and DSC*. 2022. [Online]. Available: <https://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-189419> (last visited: Jun. 9, 2023).
- [GGR19] A. I. Gomez, D. Gomez-Perez, and G. Renault, «A probabilistic analysis on a lattice attack against DSA», *Designs, codes, and cryptography*, vol. 87, no. 11, pp. 2469–2488, 2019, Place: New York Publisher: Springer US.
- [GK19] A. Goudossis and S. K. Katsikas, «Towards a secure automatic identification system (AIS)», *Journal of Marine Science and Technology*, vol. 24, no. 2, pp. 410–423, Jun. 1, 2019. [Online]. Available: <https://doi.org/10.1007/s00773-018-0561-3> (last visited: Jan. 17, 2023).
- [GNU22] GNU Radio. «GNU radio». (Mar. 2022), [Online]. Available: https://wiki.gnuradio.org/index.php/Main_Page (last visited: Jun. 5, 2023).
- [Gre] Great Scott Gadgets. «HackRF one - great scott gadgets». (), [Online]. Available: <https://greatscottgadgets.com/hackrf/one/> (last visited: Jun. 5, 2023).

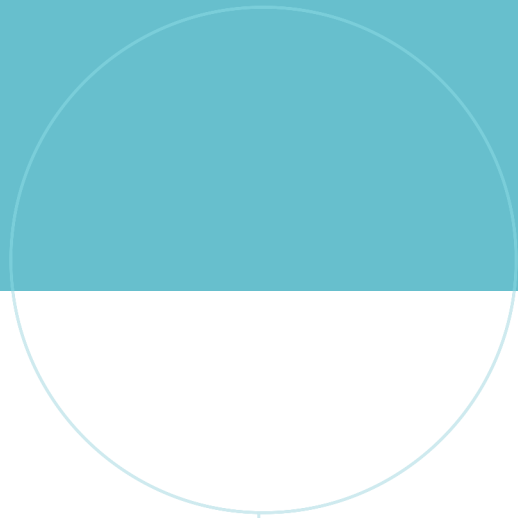
- [Hag20] K. Haga, «Breaking the cyber kill chain by modelling resource costs», Accepted: 2021-09-15T16:08:35Z, Master thesis, NTNU, 2020. [Online]. Available: <https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2777681> (last visited: Apr. 18, 2023).
- [HIB+07] H.-C. Hung, Y. Iizuka, *et al.*, «Ancient jades map 3,000 years of prehistoric exchange in southeast asia», *Proceedings of the National Academy of Sciences*, vol. 104, no. 50, pp. 19 745–19 750, Dec. 11, 2007, Publisher: Proceedings of the National Academy of Sciences. [Online]. Available: <https://www.pnas.org/doi/abs/10.1073/pnas.0707304104> (last visited: Jun. 1, 2023).
- [IAL11] IALA. «R0144 harmonized implementation of application specific messages (ASM)», IALA AISM. (Jun. 1, 2011), [Online]. Available: <https://www.iala-aism.org/product/r0144/> (last visited: Jan. 31, 2023).
- [IAL19] IALA, *THE TECHNICAL SPECIFICATION OF VDES*, Jun. 2019.
- [IAL20a] IALA, *G1143 unique identifiers for maritime resources*, Dec. 2020. [Online]. Available: <https://www.iala-aism.org/product/g1143/> (last visited: May 30, 2023).
- [IAL20b] IALA, *G1157 web service based s-100 data exchange - IALA AISM*, Dec. 2020. [Online]. Available: <https://www.iala-aism.org/product/g1157/> (last visited: May 31, 2023).
- [IAL22] IALA, *G1117 VHF data exchange system (VDES) overview*, Dec. 2022. [Online]. Available: <https://www.iala-aism.org/product/g1117/> (last visited: Feb. 20, 2023).
- [IHO19] IHO (S-100WG), *Under keel clearance management product specification*, version 4.0.0, Jun. 28, 2019. [Online]. Available: https://registry.iho.int/productspec/view.do?idx=176&product_ID=S-129&statusS=5&domainS=ALL&category=product_ID&searchValue= (last visited: May 1, 2023).
- [IHO20] IHO, *IHO data protection scheme*, version 1.2.1, Mar. 2020. [Online]. Available: <https://metanorma.github.io/mn-samples-iho/documents/s63/document.html#toc0> (last visited: May 1, 2023).
- [IHO22] IHO, *S-100 universal hydrographic data model*, version 5.0.0, Dec. 16, 2022. [Online]. Available: https://registry.iho.int/productspec/view.do?idx=194&product_ID=S-100&statusS=5&domainS=ALL&category=product_ID&searchValue= (last visited: May 1, 2023).
- [IHO23a] IHO. «Product specification register», registry.iho.int. (Jun. 9, 2023), [Online]. Available: <https://registry.iho.int/productspec/list.do> (last visited: Jun. 9, 2023).
- [IHO23b] IHO. «S-100 based product specifications | IHO». (Oct. 24, 2023), [Online]. Available: <https://iho.int/en/s-100-based-product-specifications> (last visited: Jun. 9, 2023).
- [IHO23c] IHO. «S-57 encoding bulletins | IHO». (Mar. 14, 2023), [Online]. Available: <https://iho.int/en/s-57-encoding-bulletins> (last visited: Jun. 9, 2023).

- [IHO23d] IHO, «SECOM overview», Jun. 16, 2023. [Online]. Available: https://iho.int/uploads/user/Inter-Regional%20Coordination/WWNWS/S-124PT/S-124%20PT1/S124PT1_2020_2.7_EN_SECOM_overview_v1.0.pdf (last visited: Jun. 16, 2023).
- [IMOa] IMO. «E-navigation». (), [Online]. Available: <https://www.imo.org/en/OurWork/Safety/Pages/eNavigation.aspx> (last visited: Jan. 17, 2023).
- [IMOb] IMO, «MSC 85 - annex 20 - strategy for the development and implementation of e-nav.pdf». [Online]. Available: <https://wwwcdn.imo.org/localresources/en/OurWork/Safety/Documents/enavigation/MSC%2085%20-%20annex%200%20-%20Strategy%20for%20the%20development%20and%20implementatio n%20of%20e-nav.pdf> (last visited: May 30, 2023).
- [IMOc] IMO, «MSC.1-CIRC.1610 - initial descriptions of maritime ServicesIn the context of e-navigation», MSC.1-CIRC.1610.
- [ITU12] ITU-R, *Interim solutions for improved efficiency in the use of the band 156-174 MHz by stations in the maritime mobile servic*, Mar. 2012.
- [ITU14] ITU-R, «Technical characteristics for an automatic identification system using time division multiple access in the VHF maritime mobile frequency band», International Telecommunication Union, 1371-5, Feb. 18, 2014. [Online]. Available: https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.1371-5-201402-I!!MSW-E.docx.
- [ITU22] ITU-R, «Technical characteristics for a VHF data exchange system in the VHF maritime mobile band», International Telecommunication Union, 2092-1, Mar. 7, 2022. [Online]. Available: https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2092-1-202202-I!!PDF-E.pdf (last visited: Feb. 5, 2023).
- [Kar22] Kartverket. «Elektroniske sjøkart (ENC)», Kartverket.no. (Jun. 1, 2022), [Online]. Available: <https://www.kartverket.no/en/at-sea/nautical-charts/elektroniske-sjokart-enc> (last visited: Jun. 9, 2023).
- [KMLS17] R. Khan, K. McLaughlin, *et al.*, «STRIDE-based threat modeling for cyber-physical systems», in *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, Sep. 2017, pp. 1–6.
- [LB22] C. Løkken and B. Bugten, «Security challenges of VDES and future VDES-based services», Department of Information Security, Communication Technology, NTNU – Norwegian University of Science, and Technology, Project report in {TTM4502}, Dec. 2022.
- [LB23] C. Løkken and B. Bugten, *Grc-ais-vdes-framebuilder*, original-date: 2023-05-16T12:45:45Z, May 19, 2023. [Online]. Available: <https://github.com/collinlokken/grc-ais-vdes-framebuilder> (last visited: May 30, 2023).

- [LHD22] B. Lin, W. Henry, and R. Dill, «Defending small satellites from malicious cybersecurity threats», in *International Conference on Cyber Warfare and Security*, Num Pages: 479-488, XIV-XVI, Reading, United Kingdom: Academic Conferences International Limited, Mar. 2022, pp. 479–488, XIV–XVI. [Online]. Available: <https://www.proquest.com/docview/2681924086/abstract/A7CDC742B9F24D50PQ/1> (last visited: Apr. 27, 2023).
- [Lit21] R. Litts, «Security improvements for the automatic identification system», *Electrical & Computer Engineering Theses & Dissertations*, Apr. 1, 2021. [Online]. Available: https://digitalcommons.odu.edu/ece_etds/225.
- [Loc23] Lockheed Martin. «Cyber kill chain®», Lockheed Martin. (Feb. 6, 2023), [Online]. Available: <https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html> (last visited: Apr. 17, 2023).
- [LRBJ] F. Lázaro, R. Raulefs, *et al.*, «VDES r-mode: Vulnerability analysis and mitigation concepts», *International Journal of Satellite Communications and Networking*, vol. n/a, n/a, _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/sat.1427>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/sat.1427> (last visited: Feb. 1, 2023).
- [Mar18] Maritime Safety Comitee, «E-NAVIGATION STRATEGY IMPLEMENTATION PLAN», International Maritime Organisation, MSC.1-Circ.1595, May 25, 2018. [Online]. Available: [https://wwwcdn.imo.org/localresources/en/OurWork/Safety/Documents/enavigation/MSC.1-Circ.1595%20-%20E-Navigation%20Strategy%20Implementation%20Plan%20-%20Update%201%20\(Secretariat\)%20\(2\).pdf](https://wwwcdn.imo.org/localresources/en/OurWork/Safety/Documents/enavigation/MSC.1-Circ.1595%20-%20E-Navigation%20Strategy%20Implementation%20Plan%20-%20Update%201%20(Secretariat)%20(2).pdf).
- [MCC20a] MCC. «Basic concept — maritime connectivity platform 0.11.0 documentation». (2020), [Online]. Available: <https://docs.maritimeconnectivity.net/en/latest/basicConcept.html> (last visited: May 30, 2023).
- [MCC20b] MCC. «Maritime identity registry (MIR) — maritime connectivity platform 0.11.0 documentation». (2020), [Online]. Available: <https://docs.maritimeconnectivity.net/en/latest/MIR.html> (last visited: May 1, 2023).
- [MCC20c] MCC. «Maritime messaging service (MMS) — maritime connectivity platform 0.11.0 documentation». (2020), [Online]. Available: <https://docs.maritimeconnectivity.net/en/latest/MMS.html> (last visited: May 30, 2023).
- [MCC20d] MCC. «Maritime service registry (MSR) — maritime connectivity platform 0.11.0 documentation». (2020), [Online]. Available: <https://docs.maritimeconnectivity.net/en/latest/MSR.html> (last visited: May 30, 2023).
- [MCC20e] MCC. «MCP instance provider — maritime connectivity platform 0.11.0 documentation». (2020), [Online]. Available: <https://docs.maritimeconnectivity.net/en/latest/MCPInstanceProvider.html#mcp-instance-provider-list> (last visited: May 3, 2023).
- [MCC20f] MCC. «Source code — maritime connectivity platform 0.11.0 documentation». (2020), [Online]. Available: <https://docs.maritimeconnectivity.net/en/latest/sourcecode.html> (last visited: Jun. 17, 2023).

- [MCP] MCP, «The maritime connectivity platform (MCP)».
- [MPM16] J. S. Molléri, K. Petersen, and E. Mendes, «Survey guidelines in software engineering: An annotated review», in *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ser. ESEM '16, New York, NY, USA: Association for Computing Machinery, Sep. 8, 2016, pp. 1–6. [Online]. Available: <https://dl.acm.org/doi/10.1145/2961111.2962619> (last visited: Jun. 14, 2023).
- [Muk23] P. Mukherjee. «How to order electronic navigation charts and keep them updated on ships?» (Sep. 19, 2023), [Online]. Available: <https://www.marineinsight.com/marine-navigation/order-electronic-navigation-charts-and-keep-them-updated/> (last visited: Jun. 9, 2023).
- [Nav21] Navelink. «Navelink-HOW-TO-hire-service-in-navelink-VIS-hotel.pdf». (2021), [Online]. Available: <https://www.navelink.org/wp-content/uploads/2021/12/Navelink-HOW-TO-Hire-service-in-Navelink-VIS-Hotel.pdf> (last visited: May 30, 2023).
- [Nav23a] Navelink. «Navelink-dev.-forum-2023-03-23.pdf». (Mar. 23, 2023), [Online]. Available: <https://www.navelink.org/wp-content/uploads/2023/03/Navelink-Dev.-Forum-2023-03-23.pdf> (last visited: May 30, 2023).
- [Nav23b] Navelink. «Navelink-dev.-forum-2023-04-27.pdf». (Apr. 27, 2023), [Online]. Available: <https://www.navelink.org/wp-content/uploads/2023/05/Navelink-Dev.-Forum-2023-04-27.pdf> (last visited: May 3, 2023).
- [NB] D. A. Nesheim and K. Bernsmed, «Secure, trustworthy and efficient information exchange – enabling added value through the maritime data space and public key infrastructure»,
- [NIS23] NIST. «Cyber threat - glossary | CSRC», Computer Security Resource Center. (Jun. 15, 2023), [Online]. Available: https://csrc.nist.gov/glossary/term/cyber_threat (last visited: Jun. 15, 2023).
- [Ope] OpenCPN. «About OpenCPN». (), [Online]. Available: <https://opencpn.org/OpenCPN/info/about.html> (last visited: Apr. 13, 2023).
- [Osm23] Osmocom, *Osmocom/gr-osmosdr*, original-date: 2014-09-23T20:38:06Z, May 18, 2023. [Online]. Available: <https://github.com/osmocom/gr-osmosdr> (last visited: Jun. 5, 2023).
- [PCTS02] A. Perrig, R. Canetti, *et al.*, «The TESLA broadcast authentication protocol», 2002. [Online]. Available: https://people.eecs.berkeley.edu/~tygar/papers/TESLA_broadcast_authentication_protocol.pdf.
- [PHT+09] T. Polk, R. Housley, *et al.*, «Elliptic curve cryptography subject public key information», Internet Engineering Task Force, Request for Comments RFC 5480, Mar. 2009, Num Pages: 20. [Online]. Available: <https://datatracker.ietf.org/doc/rfc5480> (last visited: Jun. 1, 2023).
- [Poo20] K. Poojara. «STRIDE-threat modeling technique | cybrary». (Oct. 15, 2020), [Online]. Available: <https://www.cybrary.it/blog/stride-threat-modeling-technique> (last visited: Jun. 16, 2023).

- [RHAL92] R. L. Rivest, M. E. Hellman, *et al.*, «Responses to NIST’s proposal», *Communications of the ACM*, vol. 35, no. 7, pp. 41–54, Jul. 1, 1992. [Online]. Available: <https://dl.acm.org/doi/10.1145/129902.129905> (last visited: May 2, 2023).
- [Skj21] S. Skjæveland. «S-100 data: What it is, how it works and why you should care». (Jun. 11, 2021), [Online]. Available: <https://blog.ecc.no/s-100-data-what-it-is-how-it-works-and-why-you-should-care> (last visited: Jun. 9, 2023).
- [Sun23] Sunshine2k. «Understanding and implementing CRC (cyclic redundancy check) calculation», Sunshine’s Homepage. (Mar. 2023), [Online]. Available: http://www.sunshine2k.de/articles/coding/crc/understanding_crc.html#ch5 (last visited: Jun. 17, 2023).
- [Suw16] S. Suwannarath, «The TESLA-alpha broadcast authentication protocol for building automation system», ISBN: 9781339744063 Publication Title: ProQuest Dissertations and Theses, M.S. California State University, Long Beach, United States – California, 2016, 122 pp. [Online]. Available: <https://www.proquest.com/docview/1797618382/abstract/77D872803E964AC1PQ/1> (last visited: May 30, 2023).
- [Voy22] M. Voytenko. «Maritime accidents monthly report SEP 2022», FleetMon.com. Section: Accidents. (Oct. 1, 2022), [Online]. Available: <https://www.fleetmon.com/maritime-news/2022/39689/maritime-accidents-monthly-report-sep-2022/> (last visited: Jan. 30, 2023).
- [Waw15] R. Wawruch, «The concept of a single window in e-navigation and according to the EU regulations», *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation*, vol. 9, no. 4, pp. 551–556, 2015. [Online]. Available: http://www.transnav.eu/Article_The_Concept_of_a_Single_Window_Wawruch,36,615.html (last visited: Feb. 1, 2023).
- [WH20] A. Walde and E. G. Hanus, «The feasibility of AIS- and GNSS-based attacks within the maritime industry», Accepted: 2021-09-23T19:06:59Z, Master thesis, NTNU, 2020. [Online]. Available: <https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2781145> (last visited: Jan. 30, 2023).
- [XYL05] L. Xu, L. Yang, and M. Lin, «Factoring large integers using parallel general number field sieve», vol. 3, Jan. 1, 2005, pp. 1017–1023.
- [AA21] AA, *Auth-AIS: Secure, flexible, and backward-compatible authentication of vessels AIS broadcasts (proof of concept)*, original-date: 2020-03-08T17:20:24Z, Sep. 11, 2021. [Online]. Available: <https://github.com/A1337CBS/Auth-AIS> (last visited: May 30, 2023).



 **NTNU**

Norwegian University of
Science and Technology