A Novel Framework to Detect Maritime GPS Spoofing Attack

MCS

By

Sehrish Batool

00000318131

Submitted to the Faculty of Department of Information Security, Military College of Signals, National University of Sciences and Technology, Rawalpindi in partial fulfillment of the requirements for the degree of MS in Information Security

JUNE 2023

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS Thesis written by **Sehrish Batool**, Registration No. **00000318131**, of **Military College of Signals** has been vetted by undersigned, found complete in all respects as per NUST Statutes/Regulations/MS Policy, is free of plagiarism, errors, and mistakes and is accepted as partial fulfillment for award of MS degree. It is further certified that necessary amendments as pointed out by GEC members and local evaluators of the scholar have also been incorporated in the said thesis.

Signature: ______________

Name of Supervisor Asst Prof Shahzaib Tahir, PhD

Date: 02-08-2023

Signature (HOD): ______________

Date: 21/8/23

Signature (Dean/Principal) ______________

Date: 21/8/23

Brig
Dean, MCS (NUST)
(Asif Masood, Phd)

DECLARATION

I certify that this research work titled “A Novel Framework to Detect Maritime GPS Spoofing Attack” is my own work. No portion of the work presented in this thesis has been submitted in support of another award or qualification either at this institution or elsewhere. The material that has been used from other sources has been properly acknowledged/referred.

____

Signature of Student
Sehrish Batool
00000318131

ABSTRACT

Navigating at sea has undergone a significant transformation, shifting from traditional navigation methods based on landmarks and celestial objects to the utilization of advanced technology such as the Global Positioning System (GPS). However, the emergence of GPS spoofing as a pervasive cyber threat poses substantial risks to maritime navigation, introducing vulnerabilities and potential disruptions to critical systems. This research aims to address the shortcomings of current defensive strategies, which are often hindered by their high costs, limited effectiveness, or impracticality in real-world maritime scenarios. Moreover, maritime GPS spoofing has been overlooked despite extensive research on general anti-spoofing techniques. To fill this research gap, a novel framework is proposed that leverages the concept of Order Preserving Encryption, a scheme that allows efficient range queries on encrypted data while maintaining the order. The framework is designed with confidentiality, integrity, and availability as primary objectives, achieved through the implementation of encryption, cloud storage, and comprehensive data comparison techniques. To evaluate the efficacy of the proposed scheme, an extensive analysis is conducted on a real-world dataset obtained from the searoutes.com API.

Keywords: Global Positioning System, Maritime GPS spoofing, Order Preserving Encryption, Cloud Computing, Searoutes.com API

DEDICATION

I would like to dedicate this thesis to my late father and my mother. My father's unwavering support, and guidance have been instrumental in shaping the person I am today. Though he is no longer with us, his memory and influence continue to inspire me in my academic pursuits. My mother's steadfast prayers have always been my source of strength.

ACKNOWLEDGEMENTS

I am thankful to the divine guidance and strength provided by Allah Almighty, enabling me to persevere and overcome challenges during the completion of this thesis within a limited timeframe. All praise and gratitude belong to HIM.

I would like to express my heartfelt gratitude to my supervisor, Dr. Shahzaib Tahir Butt, for his support, staunch guidance, and continuous encouragement throughout this transformative journey. Working under his supervision has been a great honor and privilege. His visionary outlook, dynamic methodology, timely response, and valuable recommendations have played a pivotal role in successfully accomplishing this arduous undertaking.

I extend my thanks to my co-supervisor, Dr. Fawad Khan, and committee members, Dr. Hassan Tahir, Dr. Mir Yasir and Maj. Zeeshan Zulkifl for their support and useful suggestions that have contributed to the improvement of this research study.

I am deeply grateful to the Military College of Signals and the National University of Sciences and Technology for providing me with the opportunity to enhance my research skills and make a positive contribution to society.

I would like to thank the organization and individuals who directly or indirectly participated in this research study, as their contributions were essential in implementing and testing the proposed idea.

Thank you all for your support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

ACRONYMS

Advanced Encryption Standard	AES
Amazon Web Services	AWS
Application Programming Interface	API
Automatic Identification System	AIS
Base Transceiver Station	BTS
Bring Your Own Encryption	BYOE
China–Pakistan Economic Corridor	CPEC
Client for URL	CURL
Closed-Circuit Television	CCTV
Cloud Server	CS
Cloud Service Provider	CSP
Confidentiality, Integrity, and Availability	CIA
Cypher Text	CT
Data Encryption Standard	DES
Electronic Chart Display and Information System	ECDIS
Emission Control Area	ECA
Frequency Modulation	FM
Global Positioning System	GPS
Google Cloud Platform	GCP
Hypertext Preprocessor	PHP
Hypertext Transfer Protocol Secure	HTTPS
Indistinguishable under Chosen Plaintext Attack	IND-CPA
Information technology	IT
Master Control Station	MCS
National Institute of Standards and Technology	NIST
Order Preserving Encryption	OPE
Plain Text	PT
Pseudorandom Noise	PRN

Public Key Infrastructure	PKI
Python Order Preserving Encryption	PYOPE
Radar Cross Section	RCS
Receiver Autonomous Integrity Monitoring	RAIM
Redundant Array of Independent Disks	RAID
Relational Database Management System	RDBMS
Ron Rivest, Adi Shamir and Leonard Adleman	RSA
Satellite Vehicle	SV
Secure Code Estimation and Replay	SCER
Spreading Code Encryption	SCE
Strengths, Weaknesses, Opportunities, and Threats	SWOT
Structured Query Language	SQL
Symmetric Encryption	SE
The International Business Machines	IBM
Total Ship Computing Environment	TSCE
Universal Serial Bus	USB
User Interface	UI
Video Surveillance Systems	VSS

Chapter 1

Introduction

1.1 Overview

Before the advent of technology, ships used to navigate in sea by staying as close to the shore as possible and follow the shoreline without getting lost. Sailors used to detect prominent landmarks in order to determine progression at sea. If sight of land was lost, the sun and the North Star would be used to ascertain the southern and northern directions during the day and night. Other methods to find way at sea were; use of major constellations, the directions that the birds flew and the fish swam, time measurement with an hourglass, use of maps, dead reckoning and the magnetic compass, etc.. However, those methods were not too accurate and were weather dependent[1][2][3].

The need for increased precision in navigation led to development of electronic navigation aids and improved technology. Today, Sextants, Chronometers and Radar navigation has been replaced by satellite navigation system or Global Positioning System[4][5][6] making sea navigation a lot easier.

GPS was initially developed by the US Department of Defense[7][8] for military use only. The purpose was to make military aircraft, ships and ground vehicles able to determine their accurate location anywhere in the world. Following an airline disaster in 1983, the GPS was declared operational for civilian use by Reagan administration[9]. Today, after few decades of its launch, GPS has become vital in maritime navigation, as it not only shows where to go but also helpful in detecting storms and severe weather before encounter. GPS technology is used by sea-depth-reading devices, satellite imaging, weather satellites which are crucial to survival at sea.

GPS comprises of 27+ navigation satellites circling around Earth[10]. GPS satellite transmits signal which carries information about location of satellite, orbit and location of other satellites in constellation including their status and time at which signal was being sent. The GPS receiver in vessel after receiving signal calculates position of vessel and sends it to control server over internet using GSM cellular network.

In order to initiate an assault, a spoofer endeavors to mislead a GPS receiver by transmitting counterfeit GPS signals that bear resemblance to authentic signals, or by transmitting genuine signals acquired from an alternative time or location[11]. Threats to GPS navigation therefore need to be mitigated.

1.2 Motivation

With the advent of advance technologies like 5G, WiFi and new generation satellites, transformation in marine infrastructure and networks can clearly be seen. Nevertheless, these advancements and easily available cheap hazardous devices have made attacker more powerful too. Today, a spoofing device of worth few dollars and open source code is sufficient to launch sophisticated spoofing attack[12]. In recent years, number of GPS spoofing incidents have been reported including mass and blatant attack on group of 20 vessels in the Black Sea (2017)[13] and spoofing at ports in The People's Republic of China in 2019[14].

The motivation behind selecting the topic stands the fact that work done on GPS spoofing covers general anti-spoofing techniques and **maritime** GPS spoofing has not received much attention. Designing a low cost framework capable of doing quick computation to detect spoofing attack before it become aggressive and make irreparable damage to assets including cargo and more importantly people in case of terrorism attack, is therefore need of hour.

1.3 Problem Statement

GPS is the backbone for navigation and is extensively being used in maritime navigation. Now a days, economy and security of countries relies heavily on the sea. The escalating dependence on navigation satellites for military and business operations renders them an enticing objective for adversaries.GPS is prone to cyber threats and the risk of external interference to maritime navigation and disruption of key systems cannot be denied. **Spoofing** is one of the major threat among many other cyber threats to GPS. Motives of spoofer may include but are not limited to illegal, unreported and unregulated fishing, robbery, terrorism etc. Hence, analysing and proposing a criteria to eliminate or lessen this attack vector in the maritime GPS navigation is crucial.

1.4 Research Objectives

Following are the main objectives of this research:

- Propose a novel framework for detection of GPS spoofing attack on vessel in shortest possible time.
- To enable vessel, interpret correct navigation data for accurate position calculation and generation of alarm in case of sensing fake navigation data.
- To test the proposed technique on open source data set and check its feasibility.

1.5 Scope

GPS signals can be manipulated in two ways; Attacker broadcasts fake GPS signals, which resemble genuine signals or broadcasts genuine signals but captured at a different time or different location. This research study proposes solution to the former one as latter requires access to real time GPS data and hardware implementation of real GPS receiver device. The proposed architecture compares encrypted routes of vessel to check the accuracy and legitimacy of route and does not consider the time factor. This research study focuses on timely detection of spoofing attack and ways to stop the spoofing attack are out of scope.

1.6 Contribution

The research work contributes in following ways:

- Develops an understanding of the GPS, its components and its working by provision of detailed literature review.
- Develops an understanding of a various attacks mounted against GPS with special emphasis on Spoofing.
- Designed and implemented a secure GPS framework to effectively counter spoofing threat, thereby denying opportunity of misuse to criminals.
- Data for sea routes was collected from a third-party API, encompassing 30 distinct routes and obtained at four different time intervals.
- Paves a way for future work with an endeavor to secure maritime GPS communication.

1.7 Significance of Research

Due to geography and geo-political location of Pakistan in South Asia, its economy and security relies heavily on the sea. After launch of Gwadar Port Projects under CPEC, cargo liners visits the port frequently. In order to have smooth operations, strategy to detect and mitigate GPS spoofing, a common cyber threat to marine industry, should be in place. Moreover, Pakistan defense forces specially Pakistan Navy, which ensures safety of sea line communication of country, can make use of anti-spoofing technique to combat one of cyber menaces to it.

Though our research is restricted to maritime application of a secure GPS system, yet the framework will be equally applicable to following areas:

- Aviation and flight operations
- Commercial GPS systems
- Logistics and fleet management systems
- Military and LEAs

1.8 Thesis Outline

The thesis is structured as follows:

- **Chapter 1** has covered the introductory part of the thesis. It highlighted the problem statement, motivation behind research, research objectives, thesis scope, significance and contribution of research.
- **Chapter 2** covers the preliminaries and gives overview of GPS, Cloud Computing and associated security threats. It showcases vessels navigation using GPS signal. It also gives insight to the Order Preserving Encryption we are using as security mechanism. Previous significant research work carried out in the field of GPS spoofing and their limitations has also been discussed in this chapter.
- **Chapter 3** presents the threat modeling. This chapter justifies how and why data of interest, system of interest, actors, and attack vectors has been identified, selected and explains need of encryption as mitigation strategies for GPS Spoofing.

- **Chapter 4** encompasses the working of proposed architecture in the form of flow charts, different phases of proposed detection strategy, design goals and pseudocode.
- **Chapter 5** covers the implementation part of research, results gathered and descriptive analysis as well as analysis in the form of graphs and charts.
- **Chapter 6** concludes the overall findings of research, limitation of study, contribution of research and recommendations for future research.

Chapter 2

Preliminaries and Literature Review

In this chapter, a comprehensive overview of the research topic is presented, encompassing basic concepts, theories, and relevant literature. The primary objective is to establish a strong foundation by providing an in-depth understanding of the research background and context. The preliminaries section will introduce fundamental concepts and theories, providing the necessary groundwork for the subsequent discussions. Additionally, the literature review will critically examine existing scholarly works, identifying gaps in knowledge and emerging trends within the field. By combining these elements, this chapter sets the stage for the research study and lays the groundwork for further investigation.

2.1 Preliminaries

In this section, we will delve into the security challenges encountered by the maritime industry, with a specific focus on the vulnerability posed by the Global Positioning System (GPS). By exploring the intricate workings of GPS technology, we aim to provide readers with a solid understanding of its capabilities and limitations. Furthermore, we emphasize the growing significance of cloud computing in modern navigation systems and shed light on the potential cyber threats that accompany this technology. Understanding the implications of cloud computing and its associated risks is crucial for developing robust and secure solutions. Additionally, we introduce the concept of encryption[15], highlighting its importance in safeguarding sensitive data during transmission and storage. Specifically, we focus on Order Preserving Encryption (OPE)[16], an encryption technique that preserves the order of data, enabling efficient range queries. Given the pivotal role of these technologies in our proposed solution, gaining a clear understanding of them is paramount for comprehending the subsequent discussions and effectively implementing the solution.

2.1.1 Security Challenges to Maritime

The maritime industry holds a prominent position globally, encompassing vast operations at sea that require unwavering commitment to safety. Safeguarding maritime operations necessitates a combination of expertise and constant vigilance. It is crucial to enhance security measures while simultaneously maintaining the efficient and timely flow of international commerce. Maritime security encompasses a broad range of measures aimed at safeguarding vessels from internal and external threats [17]. Certain activities present a direct and immediate threat to the shipping community, while others contribute to a broader maritime landscape that fosters a sense of insecurity and vulnerability [18]. The scope of protection extends to various areas which are discussed below:

2.1.1.1 Piracy

Maritime piracy poses a significant threat to vessels, crew, and cargo, particularly in areas known for piracy activities such as the Gulf of Aden, the Malacca Strait, and the coast of Somalia. Pirates target ships for robbery, hijacking, or ransom demands. Although piracy may evoke images of the past, the reality is that modern-day piracy continues to be a significant threat to ships carrying valuable cargo. Despite advancements in maritime security, criminals are enticed by the prospect of targeting large vessels transporting millions of dollars' worth of goods. Today's pirates are often highly organized, well-equipped, and adept at utilizing advanced communication systems and technology to carry out their illegal activities.

2.1.1.2 Terrorism

Maritime terrorism involves the use of violence or threats to intimidate or cause harm to vessels, ports, or coastal infrastructure. It can include acts of sabotage, hijackings, or attacks on critical maritime targets. The advent of modern telecommunications and international commercial logistics has expanded the scope and opportunities available to terrorists. In some cases, criminals exploit marine shipping channels as a means to transport hazardous weapons and materials.

Terrorist groups leverage transportation networks due to their ability to facilitate the movement of goods and individuals in furtherance of their objectives, making the marine shipping industry a primary target. By targeting the shipping sector, terrorists aim to undermine global political and economic stability, as well as endanger the safety of citizens.

2.1.1.3 Illegal Trafficking

The maritime domain is often exploited for the illegal trafficking of goods, including drugs, weapons, contraband, and even human trafficking. Criminal networks use ships and maritime routes for smuggling activities, posing risks to security and stability. Smuggling is not limited to the shipping industry alone; criminals frequently employ various types of vessels to illicitly transport contraband across international borders.

The vast oceans serve as both vital transportation routes for legitimate goods and avenues for the illegal import and export of prohibited items. Organized crime syndicates and international criminal networks exploit the shipping industry for the large-scale transportation of their illicit merchandise, as evidenced by numerous significant drug interdictions over the years. These criminal operations extend beyond drug smuggling and may involve the clandestine transportation of firearms and other illicit technologies that command high prices on the black market.

2.1.1.4 Illegal Fishing

Unregulated and illegal fishing practices pose significant ecological and economic challenges. These activities not only deplete marine resources but also disrupt ecosystems and harm local fishing communities. Combating illegal fishing requires robust enforcement and surveillance measures.

2.1.1.5 Cybersecurity Challenges

The widespread integration of automated IT systems in contemporary ships presents novel avenues for threat actors and hackers to execute various cyberattacks, posing significant risks of catastrophic incidents and substantial safety ramifications[19][20]. The research community has dedicated significant efforts to extensively explore vulnerabilities within the modern maritime industry[21][22][23]. The main drivers behind these assaults typically involve acquiring unauthorized remote command over ships and maritime craft, extracting valuable and confidential data for subsequent malicious activities, or deliberately disrupting the operations of the ship by compromising critical components and rendering automated systems inaccessible. Various types of cyber threats and attacks that pose significant risks to the technologically advanced vessels are as follows:

A. IT Network Systems

The maritime industry relies on various networks to transmit and process data collected by networked information systems. These networks, such as TSCE, RICE 10, SHIPNET, C3I system, SHIP system 2000, SAFENET and Smart Ship are susceptible to security vulnerabilities due to inadequate attention given to encryption and authentication techniques in their design and configuration. Outmoded and potentially vulnerable systems are accessible on the Internet, posing risks to maritime operations. The integration of shipboard information technology systems with onshore facilities increases the potential for ongoing threats. While the modern shipping industry necessitates IT systems and network connectivity for financial, legal, and remote monitoring purposes, these systems also expand the attack surface for potential hackers.

B. Automatic Identification System

AIS transponders, which are used for communication in the maritime industry, lack authentication and integrity checks, creating a vulnerability that hackers can exploit to disseminate false messages. Perpetrators have been observed employing software-defined radio to produce counterfeit "person-in-the-water" signals, resulting in ships going undetected and disseminating forged weather updates. Dependent on conceivably imprecise information can result in suboptimal judgment and disastrous consequences.

C. Electronic Chart Display Information System

Numerous studies [24][25][26][27] have extensively examined security concerns associated with ECDIS. These studies have revealed a multitude of vulnerabilities in the implementation of ECDIS software. One notable issue is that the system is often operated on outdated computers that lack security updates. Additionally, the process of downloading maps from the Internet or manually uploading them via USB introduces potential risks, as it can compromise the system during map updates [26]. In a particular study, the researchers investigated ECDIS software and revealed multiple security vulnerabilities that could empower an assailant to eliminate or reinstall system files and introduce malevolent content. Such compromises could lead to the transmission of manipulated sensor data to ECDIS, influencing navigation decisions and potentially resulting in collisions[26][27].

D. RADAR

While radar (RAdio Detection And Ranging) signals are generally insubordinate to disruptions compared to satellite signals, they remain vulnerable to interference and Distributed Denial of Service (DDoS) attacks launched by cybercriminals. During a cyber assault, radar systems are susceptible to compromise, resulting in the production of counterfeit echoes induced by external radar signals. These false echoes can provide inaccurate information about nearby objects, posing a significant risk of ship collision accidents.

E. Video Monitoring Systems

Video Monitoring Systems (VMSs) have a vital function in guaranteeing the security and well-being of maritime vessels, payload, and crew across different categories of contemporary ships. These systems are predominantly employed to oversee and trace crucial ship activities and to protect against potential risks posed by pirates and terrorists.[28]. Nevertheless, recent findings have exposed weaknesses in VMSs, giving rise to diverse security issues. For instance, Bitdefender researchers uncovered buffer overflow vulnerabilities in two prevalent CCTV camera models employed in contemporary maritime vessels. Exploiting these vulnerabilities allowed the researchers to track the compromised camera's activities and overwrite passwords. Furthermore, these vulnerabilities could potentially lead to system crashes within the VSS or serve as entry points for other cyberattacks, further compromising the system's integrity [29].

F. Global Position System

GPS and navigational technologies in the maritime industry are targeted by cyberattacks seeking to exploit design flaws and disrupt critical services. These attacks pose medium to high risks as they can result in data breaches and physical damage. Incidents have been reported where spoofed GPS signals were used to alter vessel routes undetected, and GPS signal jamming caused widespread disruption to ship navigation.

In this research, our primary focus is on the specific threat of GPS spoofing attacks in the maritime sector. By understanding the nature of these attacks and their potential impact on navigation and safety, we aim to develop effective strategy to mitigate the risk associated with GPS spoofing in the maritime industry.

2.1.2 Global Positioning System

A system that can be used to determine the location of an object on Earth is called a Global Positioning System (GPS), which consists of a network of satellites and receiving devices. Originally known as Navstar GPS, it is now owned by the US government and operated by the United States Space Force.

GPS was initially developed by the US Department of Defense for military use only. The purpose was to make military aircraft, ships and ground vehicles able to determine their accurate location anywhere in the world. Following an airline disaster in 1983, the GPS was declared operational for civilian use by Reagan administration. Today, after few decades of its launch, GPS has become vital in maritime navigation, as it not only shows where to go but also helpful in detecting storms and severe weather before encounter. GPS technology is used by sea-depth-reading devices, satellite imaging, weather satellites which are crucial to survival at sea.

GPS comprises of 27+ navigation satellites circling around Earth. GPS satellite transmits signal which carries information about location of satellite, orbit and location of other satellites in constellation including their status and time at which signal was being sent. The GPS receiver in vessel after receiving signal calculates position of vessel and sends it to control server over internet using GSM cellular network.

The GPS is made up of three primary segments, namely Space segment, Control segment, and User segment as shown in Figure 2.1. The U.S. Space Force is responsible for developing, maintaining, and operating the Space and Control segments of the GPS. The GPS satellites broadcast signals from space, which can be received by GPS receivers on the ground or in the air. The receivers use these signals to determine their three-dimensional location, as well as the current time, through a process called trilateration. This process involves measuring the time it takes for the GPS signals to travel from the satellites to the receiver, and using that information to calculate the distance between the receiver and each satellite. By combining this information from multiple satellites, the receiver can determine its precise location on Earth.

2.1.2.1 Space Segment

The Space Segment consists of 24 satellites orbiting the Earth at an altitude of approximately 20,200 km and completing one orbit every 12 hours. These satellites are distributed across six equally-spaced orbital planes, with each plane containing nominally four satellites. This segment is designed to ensure that there are always a minimum of four satellites visible above a 15 degree mask at any point on the surface of the Earth, at any given time, and in any weather condition. Each satellite in the GPS constellation is equipped with a highly accurate atomic clock, which operates at a frequency of 10.23 MHz.

2.1.2.2 Control Segment

The Control Segment assumes the duty of overseeing and upholding the GPS satellites. This encompasses monitoring the satellite's broadcasts, timekeeping, operational condition, and orbital paths. Constantly tracking the GPS satellites are five monitoring ground stations located in Kwajalein, Ascension Island, Colorado Springs, Diego Garcia, and Hawaii. In addition to ground stations, control segment is composed of a master control station (MCS), an alternative master control station and four dedicated ground antennas.

Calculating the actual orbital position of each satellite is crucial for predicting its path accurately around the clock, 24 hours a day, 7 days a week. GPS receivers detect and process the signals generated by the GPS satellite onboard sensors, allowing them to determine the precise position of each satellite. The signals transmitted by the GPS satellites are received and recorded by the ground control stations, which then analyze and estimate the measurement errors in the signals. After estimating the measurement errors, the control stations transmit this information to the Master Control Station (MCS) located at Schriever Air Force Base in Colorado Springs, Colorado. The MCS is responsible for managing and maintaining the overall accuracy of the GPS system. It uses the information received from the control stations to precisely track the positions and orbits of the satellites, as well as monitor their health and performance. The MCS can make adjustments to the satellites' orbits and clocks in real-time. Once the MCS has made the necessary adjustments to the satellites' orbits and clocks, it sends updated information to the satellites via ground antennas. The satellites then use this updated information to improve the accuracy of the signals they transmit to GPS receivers on Earth.

2.1.2.3 User/Receiver Segment

The user segment is the final component of the GPS system, and it is responsible for receiving and processing the signals generated by the satellites. All GPS users who employ GPS equipment, including a GPS navigator or receiver with an integrated microcomputer and GPS antenna, fall under this segment. Through the evaluation of signal travel time from GPS satellites to the receiver, the GPS receiver can compute its location using triangulation. By leveraging this method, the receiver can ascertain its latitude, longitude, and elevation coordinates within the WGS84 reference system. The receiver accomplishes this by using the traveling time of signals from at least four satellites. The design of GPS receivers can vary depending on the specific needs and budget of the user.

Currently, over one billion mobile devices such as smartphones, tablets, and cameras, which have GPS capabilities, have been activated. Transportation, Defense application (Missile guidance), Mapping, Geodetic control survey, Agriculture, Natural resource management, Cadastral survey, Marine/Aerial/Land navigation are some application areas of GPS technology.

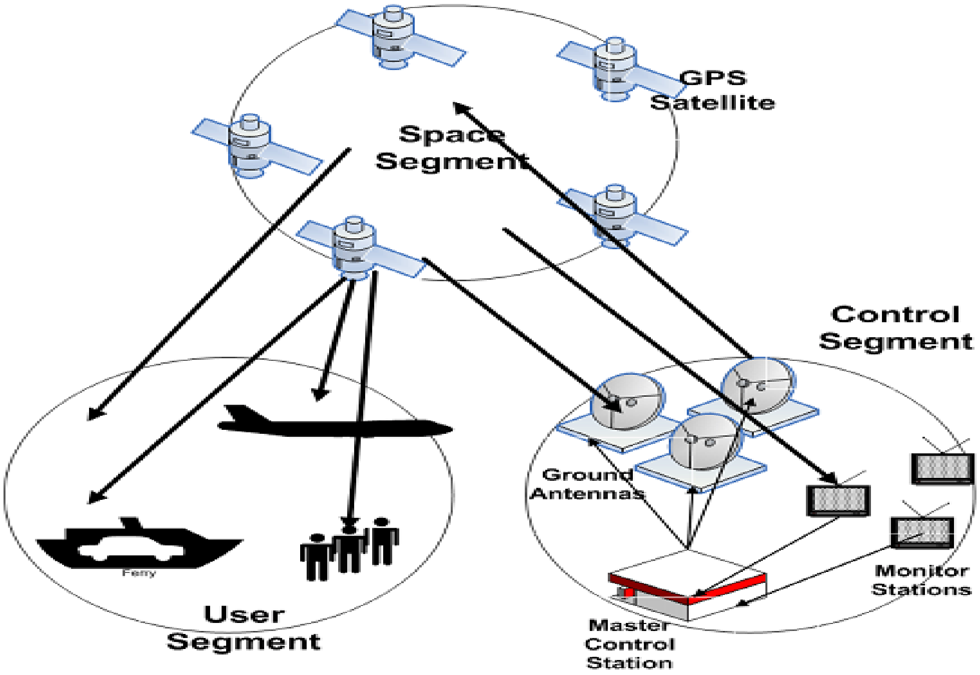


Figure 2.1 Different Segments of GPS [30]

2.1.3 GPS Signal

To use GPS effectively, one needs to have knowledge about the GPS signal structure and the methods to take measurements. Three main components of GPS signal are; Carrier Wave, Ranging Code(PRN) and Navigation Data. The GPS signal structure is summarized as below:

2.1.3.1 Carrier Wave

The GPS system functions through two carrier frequencies - the L1 carrier at 1,575.42 MHz and the L2 carrier at 1,227.60 MHz. These frequencies correspond to carrier wavelengths of roughly 19 cm and 24.4 cm, respectively, and are generated based on the speed of light in the expanse of space. The deployment of dual carrier frequencies allows for the rectification of ionospheric delay discrepancies. While all GPS satellites transmit identical carrier frequencies, each satellite utilizes a distinct code modulation to mitigate signal interference.

2.1.3.2 P-code and C/A code

The GPS system employs two distinct codes: the coarse acquisition (C/A-code) and precision (P-code). These codes consist of sequences of binary elements, commonly known as bits or chips. Referred to as PRN codes, they possess a noise-like appearance despite being generated through mathematical algorithms. Presently, the C/A-code is solely modulated onto the L1 carrier frequency, while the P-code is modulated onto both the L1 and L2 carrier frequencies. Biphase modulation serves as the underlying technique, causing a 180° phase shift in the carrier wave during transitions between 0 and 1 in the code. The C/A-code spans 1,023 binary digits, while the P-code comprises an extensive sequence that requires an extensive duration to repeat—specifically, 266 days.

2.1.3.3 Navigational Data

The GPS positioning broadcast is a data stream of low-rate information incorporated into both the L1 and L2 carriers using binary biphase modulation. It is conveyed at a speed of 50 kbps and encompasses 25 frames, with each frame comprising 1,500 bits, summing up to a total of 37,500 bits. The entire transmission of the positioning broadcast spans a duration of 12.5 minutes or 750 seconds. Within the navigation message, diverse data is conveyed, encompassing the temporal association of GPS satellites' coordinates, satellite well-being assessment, correction of satellite clocks, satellite almanac, and atmospheric data. Additionally, each satellite autonomously transmits its individual navigation message, encompassing particulars regarding the approximate positions and well-being of other satellites. Figure 2.2 visually depicts the process of generating GPS signals.

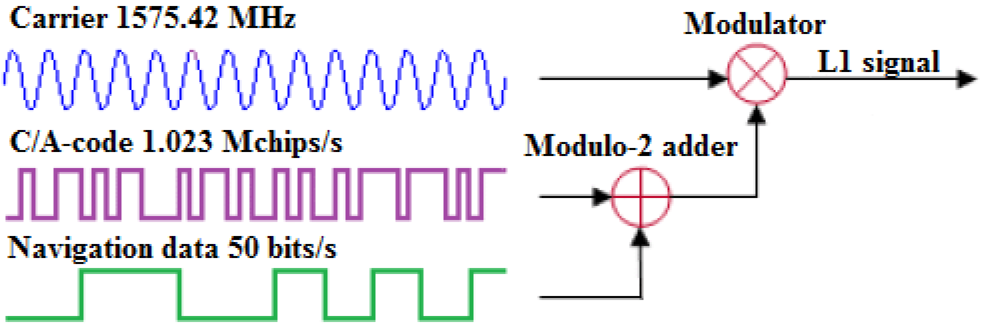


Figure 2.2 GPS Signal Generation [31]

2.1.4 Cloud Computing

Cloud computing enables users to access computing resources on-demand through a network, which includes a shared pool of configurable resources such as computing applications, platforms, software services, virtual servers, network resources , and computing infrastructure.

2.1.4.1 Cloud Computing Services

Cloud computing services can be broadly categorized into three types:

2.1.4.1.1 Infrastructure as a Service (IaaS):

This type of cloud computing provides users with virtualized computing resources such as virtual machines, storage, and networking. Users can configure and manage these resources as they see fit and can scale them up or down as their needs change.

2.1.4.1.2 Platform as a Service (PaaS):

In this model, users are provided with a platform that allows them to develop, run, and manage their own applications without having to worry about the underlying infrastructure. The platform may include programming languages, libraries, tools, and other resources necessary for application development and deployment.

2.1.4.1.3 Software as a Service (SaaS):

This form of cloud computing grants users' internet-based access to software applications, eliminating the need for local installation or maintenance. The applications are typically hosted and maintained by the service provider, who is responsible for ensuring their availability, security, and performance. Users can access the applications using a web browser or a specialized client.

2.1.4.2 Cloud Computing Deployment Models

The goal of cloud computing is to provide a flexible and scalable way to access computing resources and IT services either to an individual organization or third-party providers or both. Following are three cloud computing deployment models:

2.1.4.2.1 Private Cloud

In a private cloud, a company provides cloud services to its own employees from its own data center. The company assumes the responsibility of establishing and upholding the foundational cloud infrastructure. By implementing a private cloud infrastructure, organizations can leverage the advantages offered by cloud computing, including adaptability and user-friendliness, while upholding the customary standards of authority and safeguarding typically associated with on-premises data centers. Depending on the internal policies, users within the organization may or may not incur charges for utilizing the services, employing an IT chargeback model. Well-regarded technologies and providers for private cloud deployments comprise OpenStack and VMware.

2.1.4.2.2 Public Cloud

The public cloud model involves the delivery of cloud services by a third-party cloud service provider (CSP) over the internet. These services are available on demand and are typically sold by the minute or hour, although long-term commitments can be made for many services. Customers are charged based on the amount of central storage, processing unit cycles, or bandwidth they use. Major public CSPs include Oracle, Amazon Web Services (AWS), Tencent, Microsoft Azure, IBM, and Google Cloud Platform (GCP).

2.1.4.2.3 Hybrid Cloud

A hybrid cloud is an IT infrastructure that combines public cloud services with a private on-premises cloud environment, allowing for automated orchestration and coordination between the two. This setup enables organizations to use the private cloud for sensitive or critical applications and the public cloud for handling workload spikes or bursts. The fundamental objective of a hybrid cloud is to establish a cohesive and expandable ecosystem that harnesses the benefits of public cloud architecture while maintaining governance over essential data.

Figure 2.3 provides a visual representation of the architecture of cloud computing. It illustrates the components and relationships within a cloud computing environment, showcasing the various layers and infrastructure involved.

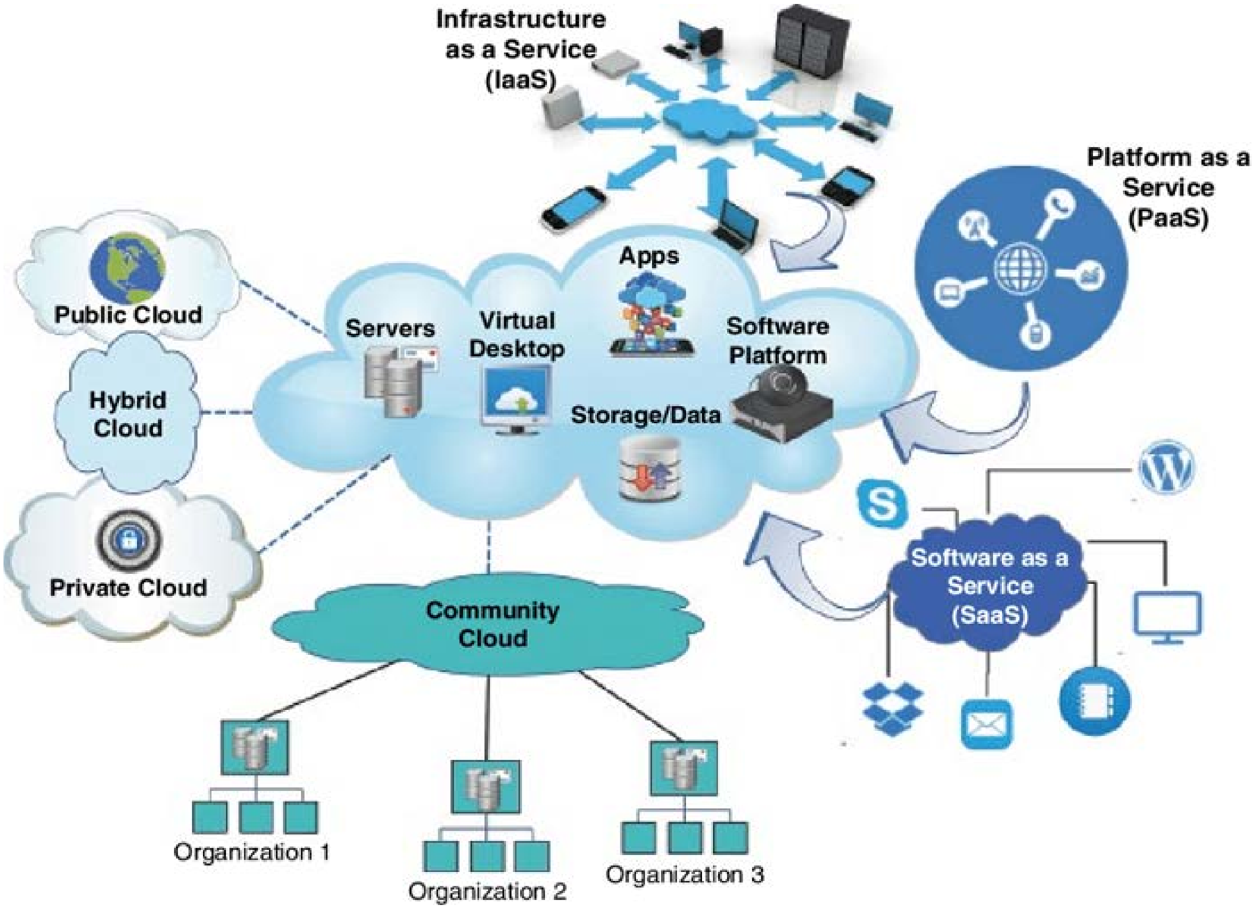


Figure 2.3 Cloud Computing Architecture [32]

2.1.5 Cloud Security

Businesses considering adopting cloud, particularly public cloud, have significant concerns regarding security. Public Cloud Service Providers (CSPs) generally allocate their hardware infrastructure across numerous customers in a multi-tenant setting, requiring significant segregation between logical computing resources. The customer's account is safeguarded with login credentials that grant access to public cloud storage and computing assets. In addition to that, researchers have explored and implemented various tools in order to establish and maintain trust in cloud environment. However, despite these efforts, there are still gaps that require attention and improvement in order to enhance the efficacy of these techniques. The primary concerns related to cloud data security encompass several aspects such as safeguarding data privacy, ensuring data protection, maintaining data availability, managing data location, and establishing secure transmission mechanisms.

2.1.5.1 Data Integrity

Maintaining data integrity in cloud systems is crucial to ensure that information is not lost or altered by unauthorized users. It serves as the foundation for cloud computing services like SaaS, PaaS, and IaaS, which require secure storage and processing of large amounts of data. Cloud computing environments often provide data processing services in addition to data storage. Data integrity can be ensured using diverse methods, including the implementation of digital signatures and the utilization of strategies akin to RAID (Redundant Array of Independent Disks).

2.1.5.2 Data Availability

Data availability refers to the ability to access and use data in a timely and reliable manner, even in the event of hardware or network failures, natural disasters, or other disruptions. It also involves ensuring that the user's data is recoverable and that the users can verify the integrity and authenticity of their data using appropriate techniques, rather than solely relying on the cloud service provider's assurances. This is important for maintaining business continuity and ensuring that critical data remains accessible and usable even in adverse conditions.

2.1.5.3 Data Privacy

In the realm of cloud computing, privacy pertains to safeguarding sensitive data and preventing unauthorized access. It is essential to ensure that cloud service providers do not expose user data to external entities. Privacy also means that users' browsing behavior and patterns should not be inferred or monitored by the cloud services or any potential attackers. This is accomplished through encryption, access control mechanisms, and other security measures.

2.1.5.4 Data Confidentiality

Maintaining data confidentiality is an essential element of data security, which guarantees the privacy and protection of sensitive information against unauthorized access. Directly storing sensitive data in cloud storage can pose risks to users, as complete trust in cloud providers may be challenging to establish, and the potential for insider threats cannot be entirely eliminated.

2.1.6 Encryption

Encryption is one of the key techniques used to achieve data confidentiality in cloud computing. It is a process of converting plain text into an unintelligible form of text called ciphertext, which can only be deciphered using a secret key or password commonly known as cryptographic key. A **cryptographic key** is basically a sequence of characters utilized in an encryption algorithm to transform data in a manner that gives it a random appearance. Similar to a physical key, it serves as a lock that encrypts data, making it accessible for decryption only by individuals possessing the correct key. Following are two primary types of Encryption:

2.1.6.1 Symmetric Encryption

In this straightforward encryption method, a single secret key is utilized for both encrypting and decrypting information. While it is the oldest and most well-known encryption technique, its primary limitation is that both parties must possess the key used for encrypting the data in order to decrypt it. Symmetric encryption algorithms, such as 3-DES(Triple Data Encryption Standard), AES(Advanced Encryption Standard) and SNOW, fall under this category. Due to its simplicity and faster execution, symmetric encryption is the preferred approach for transmitting bulk data.

2.1.6.2 Asymmetric Encryption

Asymmetric encryption, also referred to as Public Key Cryptography, is a more recent method that employs two distinct yet interconnected keys for encryption and decryption of data. One key is kept private, while the other is made public. The public key is employed for data encryption, while the private key is utilized for deciphering (and vice versa). The security of the public key is not critical since it can be freely shared over the internet. Asymmetric encryption provides a more robust approach to ensuring the security of information transmitted over the internet. RSA(Rivest-Shamir-Adleman) and Elliptic Curve Cryptography are examples of asymmetric encryption.

2.1.6.3 Cloud Encryption

Cloud encryption refers to the procedure of encoding and converting data prior to its transmission to the cloud. This process utilizes mathematical algorithms to convert plaintext data, such as text, files, code, or images, into an unreadable form called ciphertext, which can protect it from unauthorized and malicious access. By encrypting data before storing it in the cloud, even if an attacker gains access to the data, they won't be able to read it without the decryption key. This helps to protect the confidentiality of the data and ensures that it is only accessible to authorized users who have the necessary credentials to decrypt the data.

Due to the increased bandwidth usage associated with encryption, several cloud providers offer limited encryption features that primarily focus on protecting specific database fields, such as account numbers and passwords. However, these basic encryption measures does not meet the security requirements in our scenario. To address this concern, Bring Your Own Encryption **(BYOE)** model has been adopted. By using this approach, data has been encrypted on client side and responsibility for managing encryption keys rests with client. It has enables user with greater control over the encryption process and data protection in the untrusted cloud environment.

2.1.7 Order Preserving Encryption

Order-preserving encryption (OPE) is a cryptographic method that allows data to be encrypted while preserving its original ordering, enabling efficient range queries on the encrypted data. Unlike other encryption techniques, OPE does not require modifications to the underlying database management system, making it suitable for our scenario where data is outsourced and there are concerns about security from weaker adversaries. With OPE, the technique used has maintained data confidentiality while still being able to perform range queries effectively, providing a solution that balanced security and functionality. OPE is a deterministic encryption scheme.

If we have two plaintext values, p1 and p2, and apply the Order Preserving Encryption function, E, to each of them, the resulting ciphertext values, c1 and c2, will maintain the same order. Specifically, if p1 is less than p2, then the corresponding ciphertext value, c1, will be less than c2. In other words, the relative ordering of the plaintext values is preserved in the encrypted domain.

$$\mathbf{p1 < p2 \Leftrightarrow c1 < c2} \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{ (Equation 2. 1)}$$

where c1 = E(p1) and c2 = E(p2).

$$\mathbf{p1 = p2 \Leftrightarrow c1 = c2} \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{ (Equation 2. 2)}$$

Equation 2.2 states that if p1 is equal to p2, then the corresponding ciphertext c1 will be equal to c2. This property highlights the deterministic nature of OPE, where equal plaintext values always produce equal ciphertext values.

2.1.8 PYOPE Library

PYOPE is an implementation of Boldyreva's symmetric encryption scheme that preserves the order of data [33]. The authors conducted a cryptographic analysis of OPE, which was originally introduced by Agrawal et al. (SIGMOD '04) [34] for enabling efficient range queries on encrypted database data. The authors demonstrated that achieving standard security notions, such as indistinguishability against chosen-plaintext attack (IND-CPA), with a practical OPE scheme is not feasible. Instead, they proposed a security notion inspired by pseudorandom functions (PRFs) that prioritizes a high level of randomness while maintaining the order constraint. They established the security of the proposed scheme based on the pseudorandomness of a underlying blockcipher. PYOPE utilizes the connection between a random order-preserving function and the hypergeometric probability distribution, employing a black-box sampling algorithm for efficient implementation.

2.2 Literature Review

An examination and evaluation of existing literature and scholarly works related to GPS Spoofing with focus on maritime has been carried out. This section thereby explores previously proposed solutions to spoofing attacks and does a comparison between available defensive strategies in terms of their cost, effectiveness, and practicality.

Julian *et al.* [35] introduced MANA (Maritime NMEA-based Anomaly detection), a system designed with the purpose of identifying GPS spoofing in the maritime sector. MANA utilizes

NMEA-0183 data and integrates some software-based techniques to enhance its detection capabilities. Through a combination of simulations and real-world experiments, they analyzed the effectiveness of their approach, and generated a dataset for evaluation purposes. They used Pairwise Distance Monitoring (PDM) technique which detects GPS spoofing by monitoring the relative positions of two receivers. It aligns the data by interpolating the state of one receiver based on the reference receiver's data. An exponential moving average reduces noise, and if the calculated distance between the receivers' positions is below a threshold, a spoofing attack is detected.

A limitation of their work is that their simulation assumes attackers only have a single antenna, while real-world attackers may possess multiple antennas. When assailants utilize multiple antennas, the efficacy of PDM decreases. Generally, PDM requires one more receiver than the number of antennas available to the attacker for effective detection.

Singh *et al.* [36] introduced a technique for detecting and mitigating spoofing attacks targeting GNSS systems in the maritime domain using genetic programming. The utilization of genetic algorithms, Receiver Autonomous Integrity Monitoring (RAIM), and inertial sensors in the proposed method allows for the identification and discrimination of spoofing scenarios amidst regular deviations. The optimization of route selection is achieved through the incorporation of AIS data and the assessment of the present system state. Simulations were conducted using MATLAB, Network Simulator (NS-3), and Sea Clutter tools, with a ship course created using AIS dataset. The approach is theoretical and real-time implementation is required for validation.

Abreu *et al.* [37] proposed a visual analytics solution that merges spatial partitioning and trip scoring to detect anomalous activities. Users can rank trips based on segments and assess the reliability of scores by considering the amount of interpolation and visual representation on the map. The approach aims to bridge the gap in anomaly detection in the maritime navigation environment. A web tool called TOST was developed to identify local anomalies in maritime traffic. The tool allows users to filter, sort, and visualize trips on a map, and identify interpolated portions. The assumption of a single normal distribution and the need for well-partitioned sub trajectories are limitations to address.

Andrej *et al.* [38] highlighted the susceptibility of GPS to spoofing, which in turn poses risks to Electronic Chart Display and Information Systems (ECDIS) and Automatic Identification Systems (AIS). Through their research, which included a SWOT analysis of AIS and an examination of various GPS spoofing incidents occurring between 2018 and 2020, they illustrated the detrimental impact of spoofing events on ship security. They specifically investigated a case study involving an AIS spoofing incident near Elba Island in late 2019. The presence of numerous false ship signals on the ECDIS screen presents a significant technical challenge and leads to a misleading situation. Within this abundance of data, individual vessels can be easily overlooked, highlighting the necessity of employing alternative navigation methods concurrently. They concluded that, with both AIS and GPS, critical for precise positioning, being vulnerable to spoofing, relying solely on ECDIS and its overlays may introduce safety hazards.

TE Humphreys *et al.* [39] demonstrated vulnerability of civil maritime transportation to fake GPS signals and proposed a detection technique. In demo spoofing attack, the ship GPS receiver reported the position commanded by the attacker. To avoid alarm, the spoofer commanded positions that were gentle deviations of just 3o from the ships true position. A detection framework based on Doppler log, GNSS measurements and gyrocompass was proposed to detect and analyze spoofing attacks on vessels. The detector grabs the essential features of the environmental disturbances like ocean currents and wind and was designed to minimize the maximum mean integrity risk IR thereby catering for false positives 2 which hinders detection of actual spoofing incident. However, the detector developed in paper remains prone to adroit spoofing attacks. It does not detect spoofing incident before perilous condition occur or attack is in its aggressive state.

Ben Farah *et al.* [40] provided a review of the current state of cyber security in the maritime industry, focusing on both in-port and on-vessel components, systems, and services. It highlights the vulnerability of the industry to cyber-attacks specially GPS spoofing. The paper emphasizes the importance of cyber security awareness, the establishment of a legal framework and updated insurance methodologies, and transparent communication of cyber security issues among stakeholders. Standardization of digital services for autonomous vessels and the development of a new security standard for the sector are identified as future challenges that need to be addressed for the industry's economic sustainability.

Dana *et al.* [41] presented Receiver Autonomous Integrity Monitoring (RAIM), a strategy against spoofing that verifies the spatial consistency of all available GPS signals and can exclude malicious satellites. RAIM capitalizes on the frequency variation caused by the Doppler Effect in genuine signals, which introduces delays in the PRN code to maintain signal lock and complicates the spoofer's ability to maintain correlation. However, the RAIM technique assumes that any spoofing attack is limited to one or two rogue satellites rather than the entire constellation. Additionally, the RAIM user lacks detailed knowledge of the performed sanity checks, making it challenging to assess the level of protection provided.

Wullems *et al.* [42] proposed encryption as strategy to mitigate GPS spoofing attack. The widely used cryptographic technique in use to encrypt GPS signals is Spreading Code Encryption (SCE). But it is used exclusively in military applications and not applicable to Civilian GPS Signal. Possibility of replay attacks like plain meaconing or SCER and billions of devices already using unencrypted civilian signals makes it difficult to practice encryption for civilian signals[43]. A variation of the SCE technique for civilian signals was proposed by Scott *et al.* [44] but regarded as impractical. For, the proposed approach demands change to the standard signal protocols which of course is not feasible. Navigation message authentication/encryption (NMA/NME) is another cryptographic technique used as countermeasure to GPS spoofing attack. Researchers proposed approaches which make use of public key infrastructure (PKI) and embeds a signed digest of the navigation message into the navigation message itself thereby authenticating GPS signal. Later researches however showed that with sophisticated and powerful spoofing equipment, Secure Code Estimation and Replay (SCER) attacks can be mounted and bypass NMA/E[43].

Montgomery *et al.* [45] introduced another method to differentiate counterfeit signals from authentic GPS signals by making use of an antenna array. The technique uses two antennas placed close to each other and determines the pointing-angle of the satellite source. As GPS signals have various pointing angles and that of spoofed ones have just one, spoofing gets noticed easily. Another variation of this technique involves deploying multiple detectors at known locations, several meters apart (20-50 meters), and comparing their navigation solutions[46]. However, implementing these solutions necessitates significant investments due to their associated costs.

Wang *et al.* [47] put forth an approach for detecting GPS spoofing using edge computing, which relies on gathering data on edge nodes and utilizing it to verify the authenticity of received GPS signals. In the event of a spoofing attack, the method involves reconstructing the lost genuine GPS signal. While defined model is low cost, it is applicable to cars only. Moreover, it uses in-vehicle network and does not consider other anomalies that can put vehicle off the correct route.

The existing literature primarily focuses on general anti-spoofing techniques, with minimal attention given to maritime GPS spoofing. Compared to other domains, there has been limited research and development dedicated to countering GPS spoofing in maritime environments. Therefore, there is a crucial need for specialized approaches and increased focus to address the unique challenges and vulnerabilities associated with maritime GPS spoofing.

Chapter 3

Maritime GPS Spoofing Threat Model

3.1 Introduction

Threat modelling generally comprises of following main steps:

- Identification of assets, actors and attackers.
- Identification of threat and vulnerabilities (attack scenarios, leakage profiling).
- Identification of attacks and their rating.
- Mitigation strategies.

As the attack and defence sides of security are constantly changing, in order to cope up with these changes, organizations continually re-evaluate and evolve their defences. NIST threat modelling methodology therefore recommends four steps given below for threat modelling:

- Identify and characterize the system
- Identify and select the attack vectors
- Characterize the security measures to counteract the attack vectors
- Analyse the threat model

NIST methodology has been followed to do threat modelling of Maritime GPS Spoofing.

3.2 Identification and Characterization of System and Data of Interest

Global maritime sector relies crucially on Global Positioning System (system of 30+ navigation satellites circling Earth) for navigation, station keeping, and surveillance. GPS satellite transmits signal which carries information about location of satellite, orbit and location of other satellites in constellation including their status and time at which signal was being sent. The GPS receiver in vessel after receiving signal calculates position of vessel and sends it to control server over internet using GSM cellular network.

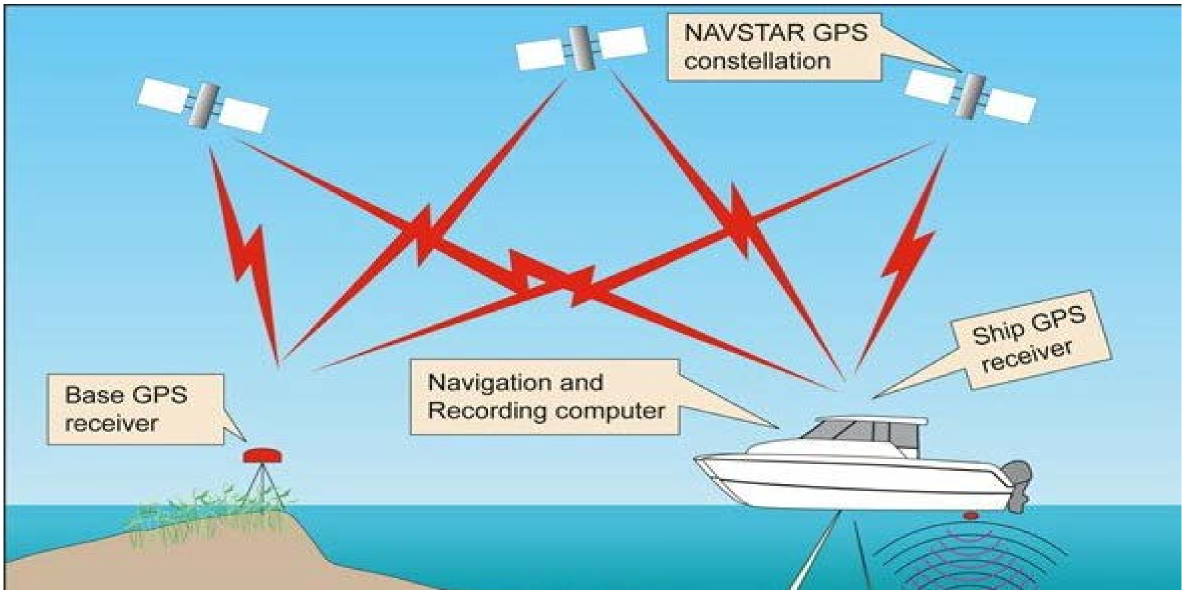


Figure 3.1 Maritime GPS Communication [48]

As demonstrated in Figure 3.1 Navstar GPS, base GPS receiver, computer and ship GPS receiver, the basic components of maritime GPS communication, work in a coordinated manner to enable accurate positioning and navigation for ships. The base GPS receiver provides a reference point, the Navstar GPS constellation provides the signals, the navigation and recording computer processes the data, and the ship GPS receiver receives and utilizes the GPS signals to determine the ship's position and support navigation tasks.

3.2.1 Actors

Actors of the system are:

- The Captain / Master
- Chief Officer
- Chief Engineer
- Personnel in charge of the daily management of maritime operational technology (OT), information technology (IT), and communication systems.

3.2.2 Attacker

In this scenario, the attacker is a spoofing entity that attempts to deceive a GPS receiver by transmitting counterfeit GPS signals that mimic genuine signals, or by transmitting authentic signals acquired from a distinct time or location. Figure 3.2 illustrates that in the presence of spoofed signals, if the ship's GPS receiver receives and relies on them, it can generate an erroneous ship position that deviates from the actual location of the ship.

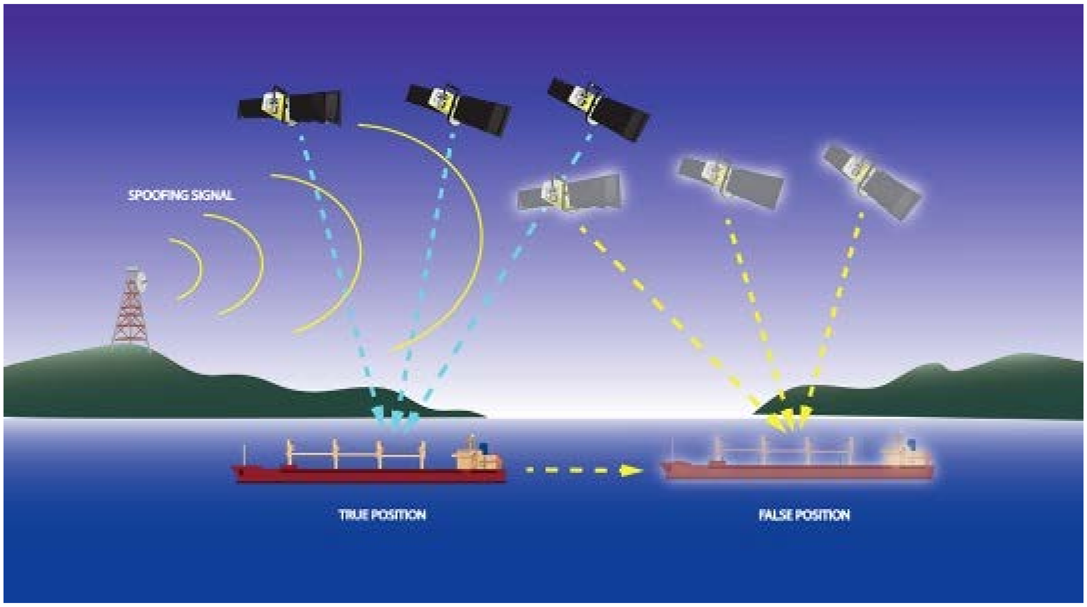


Figure 3.2 Maritime GPS Spoofing Attack [49]

3.2.3 Data of Interest

Data of interest is GPS signal. The GPS signal is basically a modulated wave which comprises of following three parts:

1) Carrier Wave, sinusoidal in nature, helps transmitting GPS signal to receiver from satellite without getting it lost in atmosphere.

2) Pseudo Random Number (PRN) code, binary in nature, holds identification information of Satellite Vehicle (SV).

3) Binary NAV data, comprising ephemeris information for satellite position calculation and almanac data indicating the temporal and status details of the complete satellite constellation, is essential for accurate navigation.

3.2.4 System of Interest

System of interest in this case is area around vessel's GPS receiver.

3.2.5 Authorized Locations

The authorized locations for the *data of interest* are as follows:

1) Storage: Signal parameters stored in GPS device or on remote server.

2) Transmission: Sent to server over wireless network.

3) Execution environment: Controller embedded inside GPS device.

4) Input: Fed in using GPS antenna.

5) Output: Displayed to the GPS Information Screen.

For data of interest, **integrity, confidentiality and availability** all are important.

3.3 Identification and Selection of Attack Vectors

Following are the ways for attacker to enter Maritime GPS system/network and selected threat vectors that are considered for the research:

3.3.1 All Attack Vectors

Location 1: Signal parameters stored in GPS device

- *Vector 1a:* Attacker gains unauthorized physical access to vessel and consequently the GPS device by breaking into physical security system.
- *Vector 1b:* Attacker replaces actual GPS receiver by fake one having fake signal parameters fed into it.

Location 2: Sent to remote server over wireless network.

- *Vector 2a:* Attacker impersonate by using false Base Transceiver Station and performs man-in-the-middle attack.
- *Vector 2b:* The attacker can send the CHANNEL REQUEST message to the Base Station Controller for several times which can lead to DoS attack.
- *Vector 2c:* The attacker can misuse the previously exchanged messages between the GPS device and server in order to perform the replay attacks.

Location 3: Controller embedded inside GPS device

- *Vector 3a:* Attacker hack device’s controller part and tamper processing of signal.

Location 4: Fed in using GPS antenna

- *Vector 4a:* Attacker uses a radio transmitter to send a counterfeit GPS signal to a GPS receiver antenna of vessel. Vessel miscalculates its position and send wrong location information to server.
- *Vector 4b:* Attacker capture and retransmits legitimate GPS signals after a delay causing the receiver to miscalculate its time.

Location 5: Displayed to the GPS Information Screen.

- *Vector 5a:* Attacker infiltrates vessels IT network and control server using spear phishing.

- *Vector 5b:* Attacker gains access to control station via compromised or misconfigured endpoints.
- *Vector 5c:* Attacker gets access to control systems and devices using stolen credentials.

3.3.2 Selected Attack Vectors

As GPS relies on signals broadcast from the satellite constellation and can be received by any GPS configured device, attack vectors relevant to it are more crucial and have adverse consequences for maritime safety and commerce if not taken care of.

Vector 4a: Vessel calculates its position using GPS signals which are weak; attacker sends fake strong signals to vessel GPS receiver antenna and make vessel believe its fake location. Attacker thus gets hostile control of vessel.

Vector 2a: Vessel calculates its location using information the GPS signal carries; attacker can perform man-in-the-middle attack by impersonating BTS and dictate vessel a false route thus luring in vessel to forbidden areas and gets control over it.

Either of the aforementioned attacks can be initiated either by a neighboring vessel tailing the target ship or through covert transmission of falsified signals from the target ship's deck.

3.4 Security Measures to Counter Attack Vectors

In order to mitigate the selected attack vectors, a cloud based solution has been proposed. A framework has been designed to achieve confidentiality, integrity and availability as major design goals with **Cloud Encryption** as core security measure.

3.4.1 Cloud Encryption

Encryption is a fundamental and effective method for safeguarding cloud data from being compromised, stolen, or read by individuals with harmful intentions. Cloud storage providers commonly employ encryption techniques to safeguard data, granting users access to encryption keys for secure decryption. These keys enable the conversion of encrypted data into its original readable form when required. Encrypted data can generally be categorized into three forms: data in transit, data at rest, and data in use.

3.4.1.1 Data in Transit

Data-in-transit, also known as "in motion", relates to data that is actively being transferred from one place to another. It is crucial to consider that data transfers may involve multiple entities, beyond just the sender and recipient. For example, transferring data within a local area network (LAN) from PC or a laptop involves a single-party data transfer. Conversely, conducting a transaction on a decentralized database such as blockchain involves data transfers among an indeterminate number of participants.

3.4.1.2 Data at Rest

Data-at-rest pertains to data that is stored or saved somewhere without being actively used or transferred to anyone or any system, including humans, third parties, or software. This category of data can be saved on different devices or platforms, including mobile devices, network-attached storage, local disk drives, database servers, USB flash drives, system directories, and other physical or virtual storage systems.

3.4.1.3 Data in Use

Data-in-use refers to the active utilization of data by one or more applications, as opposed to its storage on external repositories or physical drives. In this state, the data is actively undergoing processes such as modification, viewing, deletion, appending, or creation. Due to its nature, data-in-use is susceptible to various threats and vulnerabilities, depending on its location within the system and who can access it. It can be challenging to encrypt data-in-use since it may cause the application that has access to it to crash.

3.4.2 Methods of Cloud Encryption

There are two commonly utilized encryption techniques, recognized as encryption algorithms, for encoding and decoding data. These approaches continuously advance as the realm of information technology adjusts to bolster data security and safeguard privacy. These techniques are:

3.4.2.1 Symmetric Algorithm

This approach employs a shared key for both encrypting and decrypting data. It is well-suited for closed environments and individual users. The keys are utilized to ensure secure communication and are commonly applied in bulk data encryption. This method can be rapidly and conveniently implemented in hardware, resulting in faster processing compared to the asymmetric method. However, it is important to note that anyone possessing the key can decipher the data, regardless of their intended recipient.

3.4.2.2 Asymmetric Algorithm

This approach employs a pair of mathematically connected keys, namely a private key and a public key, which are distinct from each other. It is referred to as asymmetric encryption because the keys are paired but not identical. The private key is required to be kept confidential and secure, while the public key can be freely shared with others.

3.4.3 CIA Attainments

The CIA triad symbolizes the core elements of information security, which are confidentiality, integrity, and availability. Confidentiality involves restricting access to data, integrity is concerned with ensuring the accuracy of data, and availability is focused on ensuring that data can be accessed by authorized parties when needed. The proposed framework accomplishes these design goals as follows:

3.4.3.1 Confidentiality

In order to achieve confidentiality as design goal, a technique called **Order Preserving Encryption (OPE)** has been used. Order-preserving **symmetric** encryption is a deterministic encryption scheme that allows efficient range queries on encrypted data. OPE allows for the assessment of ciphertext values to determine the associated correlation between the initial plaintexts. It makes efficient inequality comparisons on the encrypted data without decrypting them. The proposed framework collects route of vessel from one port to another, encrypt it and store that encrypted data on cloud. Once vessel starts its journey, actual GPS coordinates are collected at random time intervals, encrypted and gets compare to the encrypted coordinates fetched from cloud. This way **confidentiality** of data, which are coordinates in our case, don’t get compromised. Coordinates are in encrypted form when they are stored at Cloud, during transmission to and from the cloud, and during comparison thereby ensuring security in all three forms: data at rest, data in transit, and data in use.

3.4.3.2 Integrity

In order to launch spoofing attack, spoofer can alter route information. However, the designed solution does exhaustive comparisons of stored and actual route vessel follows to detect tampering of data, enabling control tower to quickly respond to the attack and find a solution.

Owing to the number of entities and access points in a cloud environment, verifying the integrity of data fetched from cloud is inevitable. The framework allows comparison of data with three different datasets and therefore even a small discrepancy can easily be detected.

3.4.3.3 Availability

Adverse weather conditions or accidents such as damage to the GPS receiver may prevent a vessel from receiving route information and result in the vessel being unable to continue its journey or following an incorrect path. The designed system stores copy of route on cloud in encrypted form which can be accessed anytime to cope up with above mentioned scenarios.

Chapter 4

Proposed Framework to Detect Maritime GPS Attack

In this chapter, the proposed framework for detecting maritime GPS spoofing attacks is presented. The framework utilizes a hybrid scheme of encryption and involves an extensive comparison process. The operation of the framework is illustrated through a flowchart and sequence diagram, depicting the various stages that the data undergoes to ensure confidentiality, integrity, and availability objectives are met.

4.1 Overview

GPS spoofing involves the imitation of satellite navigation signals by an entity or individual, transmitting an indistinguishable signal of significant magnitude to overpower the genuine transmission. Since GPS receivers generally lock onto the most potent signal, the presence of the counterfeit signal can mislead onboard navigation systems, leading to erroneous computations related to vital aspects such as speed, position, and direction.[50].

The proposed framework aims to identify inconsistencies in the signal by following these steps:

1. Collecting all the coordinates of the route that the vessel is supposed to take using a third party API.
2. Encrypting the collected coordinates and sending them to the cloud, referred to as $\mathbf{Nav_{cld}}$.
3. When the vessel begins its route, the application enables the user to issue a command to compare a batch of data.
4. The application encrypts the current batch of GPS navigational data, referred to as $\mathbf{Nav_{sat}}$.
5. The corresponding batch of data is fetched from the cloud.
6. The application performs the comparison, taking into account the noise (represented by $\boldsymbol{\eta}_{\mathbf{env}}$) caused by real environmental factors such as wind and current.
7. The comparison results are displayed.

$$\mathbf{Nav_{sat} - Nav_{cld} = 0 + \eta_{env}} \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{(Equation 4. 1)}$$

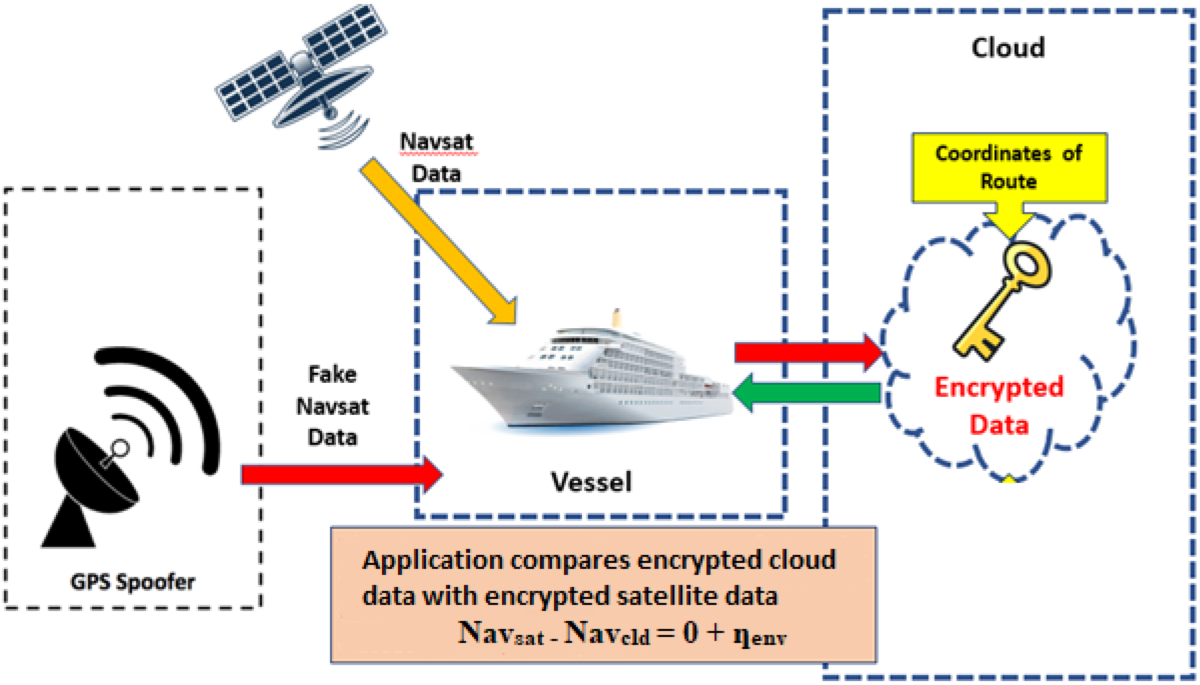

Cloud
Navsat
Data
Coordinates of
Route
Encrypted
Data
Fake
Navsat
Data
Vessel
GPS Spoofer
Application compares encrypted cloud
data with encrypted satellite data
Navsat - Navcld = 0 + ηenv

Figure 4.1 Overview of Proposed Scheme

4.2 Assumptions

To facilitate a better understanding of the proposed technique and the scope of the study, the following assumptions are made:

1) Data obtained from the API providers is accurate and legitimate.
2) The encryption technique employed adheres to established encryption standards, and the management of private keys is entrusted to a trusted entity.
3) A reliable connection to the Cloud is available and that the data stored in the Cloud is accessible.
4) Client side/vessel is safe from Insider attacks.
5) Attacker does not have physical access to vessel.

4.3 Phases of Proposed Technique

The key components of the suggested method are depicted in Figure 4.2, accompanied by their respective explanations outlined below:

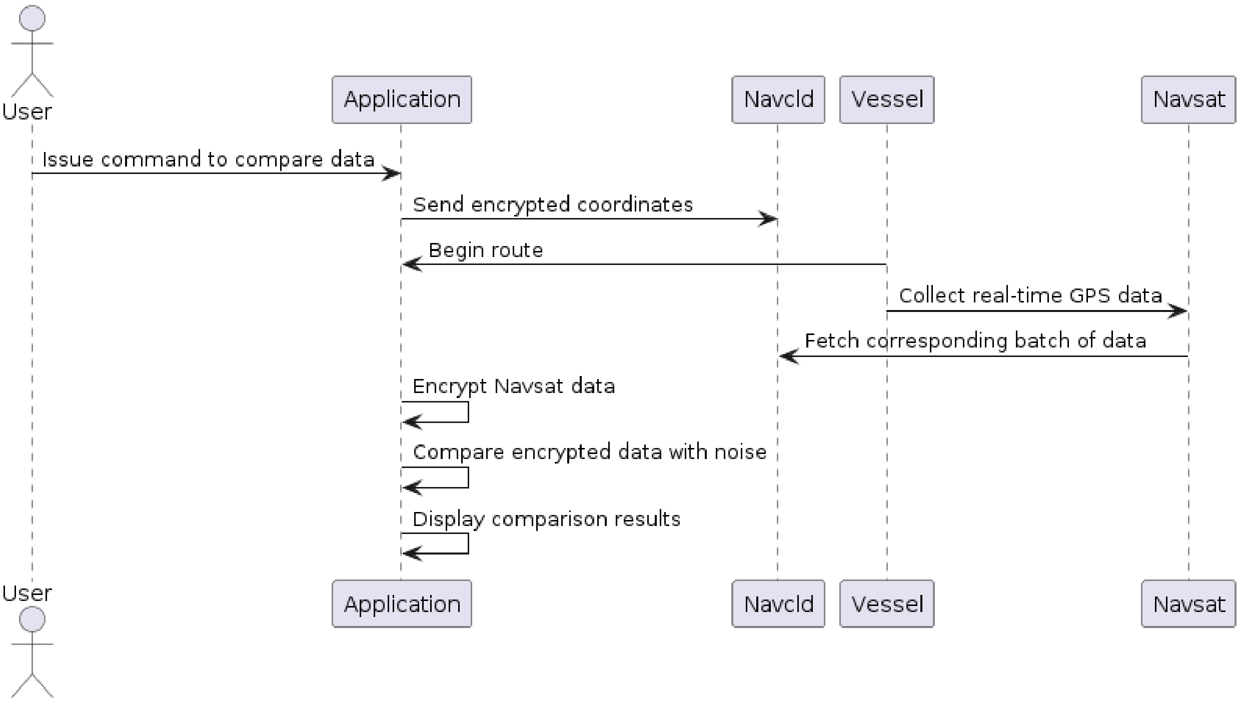

User
Application
Navcld
Vessel
Navsat
Issue command to compare data
Send encrypted coordinates
Begin route
Collect real-time GPS data
Fetch corresponding batch of data
Encrypt Navsat data
Compare encrypted data with noise
Display comparison results
User
Application
Navcld
Vessel
Navsat

Figure 4.2 Sequence Diagram of Proposed Technique

4.3.1 Reference Data Collection

In this phase, the precise and accurate coordinates of the intended route that the vessel is scheduled to navigate gets collected using a third party API searoutes.com[51]. Their web service provides users with the capability to programmatically interact with a range of tools and services offered within the Searoutes ecosystem. The sea route between a specified source location and target location, including information such as the route distance (in meters), duration (in milliseconds), and the areas crossed along the route is retrieved in this step. This phase is of significant importance as the data collected during this phase serves as a reference.

4.3.2 Encryption

This phase employ robust and industry-standard encryption algorithm to transform the collected coordinates into a secure and unreadable format. This ensures that the tentative vessel route remains confidential and protected from unauthorized access when it is at rest, in transit or in use.

Plain Sea route
PYOPE Encryption
Encrypted Sea Route

Figure 4.3 PYOPE Encryption Phase

4.3.3 Cloud Data Storage

At this stage, a connection is established to the cloud-based storage system, and encrypted coordinates gets transmitted for reliable and centralized storage. Data transferred and stored to cloud is referred as $\mathrm{Nav}_{\mathrm{cld}}$. This procedure guarantees the integrity and availability of the data while adhering to strict security protocols.

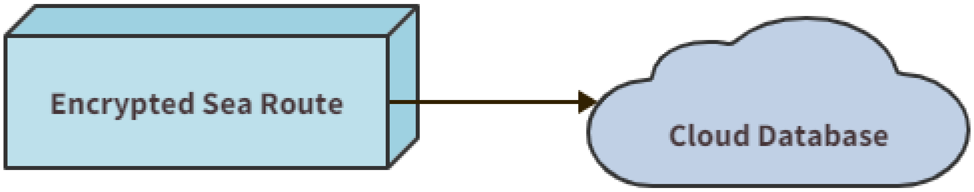


Figure 4.4 Cloud Storage Phase

4.3.4 User Command

This stage provide user with an intuitive and user-friendly interface that allows him to issue specific commands and instructions to initiate the desired operations, such as data comparison. User can do it at any point of time throughout the journey of vessel. The application enables user to enter locode of source and destination ports, and batch of data he wants to do comparison for.

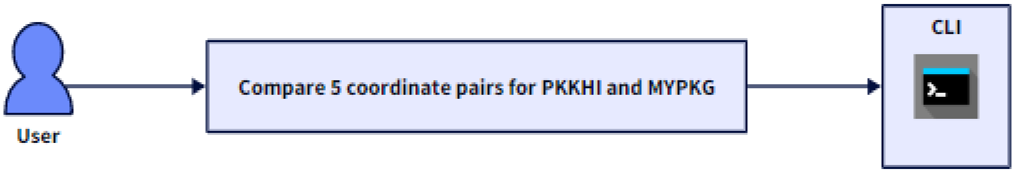


Figure 4. 5 User Command Phase

4.3.5 GPS Data Encryption

This step implements encryption of the batch of GPS navigational data using the identical encryption mechanisms employed in phase 2. Resultant data is denoted as Nav_{sat}. This process again ensures that the sensitive information is shielded from interception or tampering, safeguarding its integrity and confidentiality. Though vessel is trusted to be safe from insider attack, encryption at this point is important to make data comparable to encrypted data at cloud.

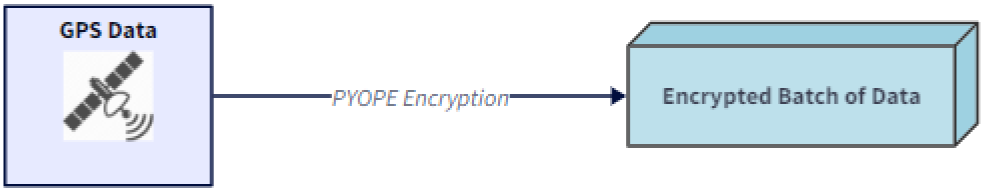


Figure 4.6 GPS Data Encryption Phase

4.3.6 Data Retrieval

During this phase, the batch of encrypted data corresponding to the GPS batch data is retrieved from the cloud storage system. Connection between Cloud and vessel is assumed to be secure which ensures the retrieval process maintains the data's integrity and authenticity.

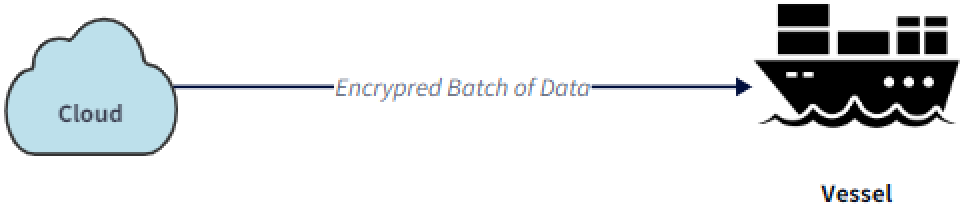


Figure 4.7 Data Retrieval Phase

4.3.7 Comparison

This phase performs a comprehensive and meticulous comparison between the encrypted Nav_{sat} data and the retrieved batch of data i.e. Nav_{cld}. During the comparison operation, consideration is given to environmental factors, noise (represented as ηenv), which can potentially impact the accuracy and reliability of the data. This comparison process facilitates the identification of any discrepancies or anomalies in route of vessel.

4.3.8 Analysis

A thorough analysis of the comparison results is conducted in this phase to extract valuable insights. This analysis enables informed decision-making processes and aids in identifying potential issues or deviations from expected outcomes. If the difference between Nav_{sat} and Nav_{cld} is determined to be zero, the vessel receives a green signal to proceed along the designated route. Conversely, if a disparity is detected between the two sets of data, a warning is displayed on the client's screen, empowering the vessel staff to initiate the required actions.

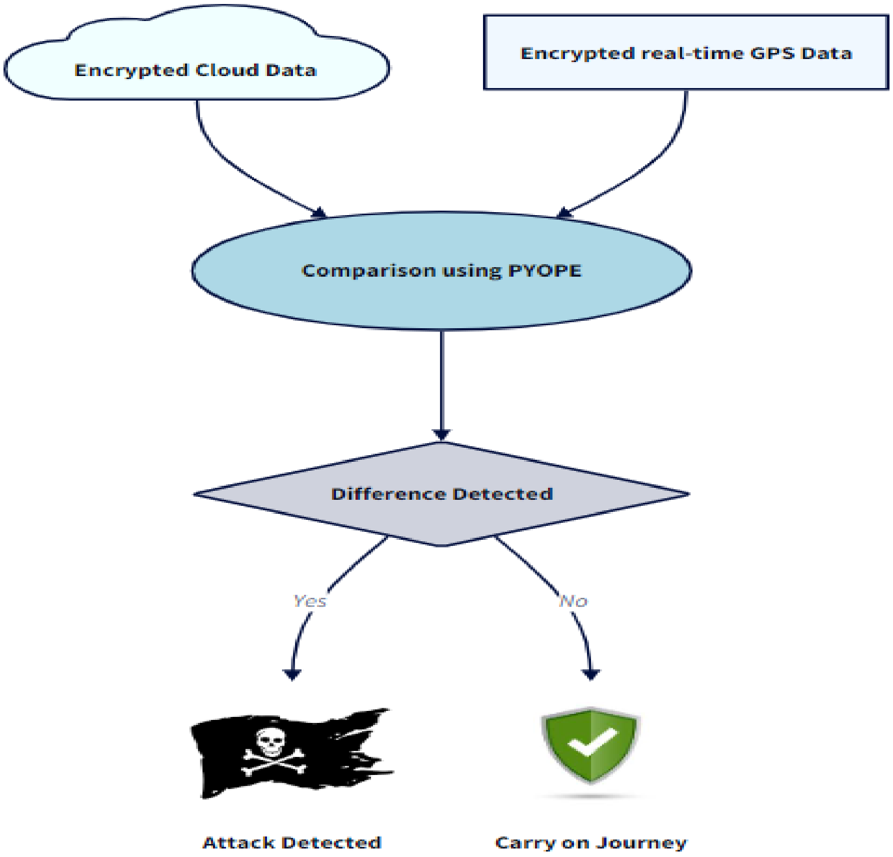


Figure 4.8 Comparison and Analysis Phase

4.4 PYOPE Algorithm

Order-preserving encryption (OPE) is a technique used to encrypt data in such a way that efficient comparisons can be made on the encrypted items without the need for decryption. This allows for quick inequality comparisons to be performed on the encrypted data. OPE not only allows for effective range queries, but also ensures that indexing and query processing can be carried out with comparable efficiency to unencrypted data. With OPE, a query involves encrypting specific values and the server can locate the corresponding ciphertexts using standard tree-based data structures in logarithmic time. This allows for precise and efficient data retrieval and processing. The PYOPE algorithm can be divided into the following key steps:

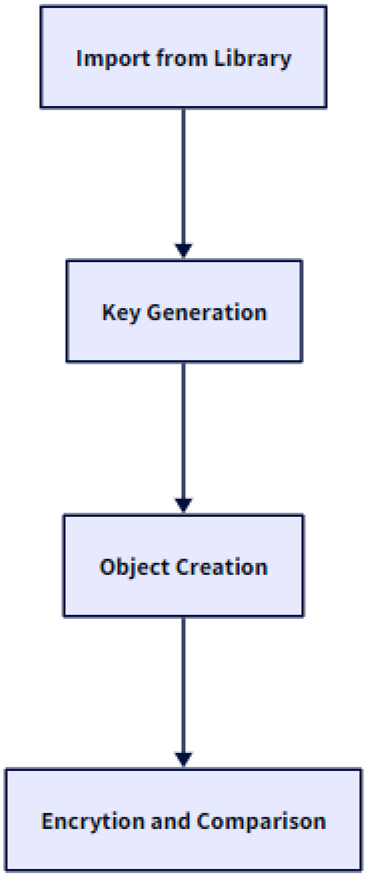


Figure 4. 9 Basic PYOPE Algorithm

Following code snippets demonstrate the usage of the OPE encryption scheme[52]:

1. Import the OPE class from the pyope.ope module:

   ```
   from pyope.ope import OPE
   ```

2. Generate a random encryption key:

   ```
   random_key = OPE.generate_key()
   ```

3. Create an instance of the OPE class using the generated key:

   ```
   cipher = OPE(random_key)
   ```

4. Perform an assertion to ensure that the encrypted value of 1000 is less than the encrypted value of 2000, which is further less than the encrypted value of 3000:

   ```
   assert cipher.encrypt(1000) < cipher.encrypt(2000) < cipher.encrypt(3000)
   ```

Formal definition of algorithm is as below:

Order-Preserving Encryption, a symmetric cryptographic technique denoted as *OPE = (K, Enc, Dec),* involves following three algorithms and operates on a ciphertext-space *R* and plaintext-space *D*.

(K) ← KeyGen(s): is the randomized key generation algorithm which generates a secret key *K* on provision of optional block size s.

(c) ← Enc(K,D,R,m): is the encryption algorithm which utilizes the secret key *K*, as well as the designated ciphertext and plaintext spaces *R* and *D*, along with a plaintext message *m* to produce a ciphertext *c*.

(m) ← Dec(K,D,R,c): is the decryption algorithm, which is deterministic, utilizes the secret key *K*, as well as the designated ciphertext and plaintext spaces *R* and *D*, along with a ciphertext *c* to return the corresponding plaintext message *m*.

Let *A* and *B* be subsets of the natural numbers *N*, with the cardinality of *A* being less than or equal to the cardinality of *B*($|A| \leq |B|$). A function f: $A \rightarrow B$ is considered *order-preserving* or strictly increasing if, for any $i, j \in A$, the condition $f(i) > f(j)$ holds if and only if $i > j$.

Another definition is, *SE* is said to be *order-preserving* if the function *Enc(K, ·),* with *K* being any key output by the key generation algorithm, is an *order-preserving function* from the plaintext-space *D* to the ciphertext-space *R*. In this case, the elements of *D* and *R* are interpreted as numbers and encoded as strings.

Correctness: An OPE scheme is correct if the following is true:

$Dec(K, D, R, (Enc(K, D, R, m)) = m$ for all *K* output by *KeyGen* and all $m \in D$

The correctness condition states that when a plaintext message "*m*" is encrypted using the encryption algorithm *Enc* with the secret key *K*, and then decrypt the resulting ciphertext using the decryption algorithm *Dec* with the same secret key *K*, the output should be equal to the original plaintext message "*m*". In other words, the encryption and decryption operations should be inverse operations, such that encrypting and then decrypting a message returns the original message.

Next, we will individually present and discuss each of the aforementioned phases in accordance with our proposed scheme. Table 4.1 illustrates the abbreviations employed in our scheme.

Enc – Represents the encryption algorithm

CS – Denotes a Cloud Server

s – Represent the block size to be considered for key genereation

M – Denotes coordinates or latitude-longitude pairs

T – Represents a local database table containing sea route coordinates

BRes – Represents result in the form of true or false

Dec – Denotes the decryption algorithm corresponding to Enc.

B – Represents batch of latitude-longitude pairs

Table 4. 1 Abbreviations used in Proposed Scheme Algorithm

A) KeyGen Phase:

The KeyGen algorithm facilitates key generation to be used for encrypting and decrypting coordinates. The algorithm takes as input a block size *s* of type integer. If not provided, default block size is 32. The algorithm returns a random key of type string.

Phase 1: KeyGen

a) Input: A block size s.

b) KeyGen: Generate random key *K* using built-in function *urandom* and *b64encode*. A block size *s* serves as an input to urandom function which outputs *random_seq*; a string containing random characters. *random_seq* is fed as an input to *base64.b64encode* function which finally generates random key *K*.

random_seq ← os.urandom(s)

K ← base64.b64encode(random_seq)

c) Output: A random key *K* for encryption

B) Enc Phase:

The encryption phase *Enc* is responsible for encrypting latitude/longitude values. It takes a randomly generated key *K*, an input range *D*, an output range *R*, and the latitude-longitude pairs *M* as inputs. To generate the input range *D*, the function *ValueRange* is invoked with the lower limit *i_low* and upper limit *i_upp* as inputs. This function returns a range of consecutive integers within the specified boundaries (*i_low* and *i_upp*, inclusive). Similarly, the output range *R* is generated using the function *ValueRange*, with the lower limit *o_low* and upper limit *o_upp* as inputs. It returns a range of consecutive integers within the specified boundaries (*o_low* and *o_upp*, inclusive). The *Enc* algorithm returns the encrypted latitude-longitude pairs *C*.

Phase 2: Enc

a) Input: A randomly generated key *K*, an input range *D*, an output range *R*, and the latitude-longitude pairs *M*.

b) Range Generation: Optional user defined upper and lower limits are passed to *ValueRange* function which generate input space *D* and output space *R*.

$D \leftarrow ValueRange(i\_low, i\_upp)$

$R \leftarrow ValueRange(o\_low, o\_upp)$

c) Initialization:

- OPE object *obj* is created by passing *K*,*D* and *R* to OPE class.
- *encypt* function is called for latitude/longitude single value *m*.

$obj \leftarrow OPE(K, D, R)$

$c \leftarrow obj.encrypt(m)$

d) Iteration: Encryption process is iterated to cater for *n* number of latitude-longitude pairs stored in table T.

for *n* in range*(size of (T))*:

$C \leftarrow obj.encrypt(n)$

e) Output: Encrypted latitude-longitude pairs *C*.

C) CS_Storage Phase:

Encrypted coordinates *C* are transferred to Cloud Server *CS* in this phase.

Phase 3: CS_Storage

a) Input: Encrypted latitude-longitude pairs *C*.

b) Output: Transmit *C* to *CS*.

D) <u>Compare_Data Phase:</u>

Coordinate pairs *c_loc* stored in local database are fetched. Encryption is carried out on *c_loc* to get *CL*; encrypted local coordinate pairs. To perform comparison, encrypted coordinates *C* are fetched from *CS*. Comparison function *Comp* takes CL and C as input, checks for discrepancy and returns boolean result *BRes*.

Phase 4: Compare_Data

a) Input: Encrypted local coordinate pairs *CL* and encrypted coordinates *C* stored on *CS*.

b) Initialization: Batch *B* of *CL* get encrypted:

```
        for c_loc in range(size of (B)):
        CL ← obj.encrypt(c_loc)
```

C is fetched from *CS* for comparison

c) Comparison:

```
        BRes ← Comp(CL,C)
```

d) Output: True or False *BRes.*

E) <u>Dec Phase:</u>

The encryption phase Dec is responsible for decrypting latitude/longitude values. It takes a randomly generated key K, an input range D, an output range R, and encrypted latitude-longitude pairs *C* as inputs. The *Dec* algorithm returns the decrypted latitude-longitude pairs *M*.

Phase 5: Dec

a) Input: Randomly generated key *K*, an input range *D*, an output range *R*, and encrypted latitude-longitude pairs *C*.

b) Decryption: To decrypt batch *B* of latitude-longitude pair *c*.

```
for c in range(size of (B)):
    M ← obj.decrypt(c)
```

c) Output: Decrypted latitude-longitude pairs *M*

By diligently following essential steps, the proposed approach ensures the robustness, confidentiality, and reliability of the collected data, allowing for accurate comparisons and insightful analysis to support critical decision-making processes in maritime operations.

Chapter 5

Implementation, Results and Analysis

This chapter provides a practical implementation of the PYOPE library, showcasing how it has been utilized to implement the proposed framework. It details the process of collecting, storing, and utilizing actual sea route data. The chapter concludes with the presentation and analysis of the obtained results.

5.1 Introduction

Instances of substantial GPS disruption have been globally documented within the maritime sector. The occurrence of such disturbances can result in the deprivation or alteration of GPS signals, exerting an influence on pivotal operations such as bridge guidance, timing derived from GPS, and communication frameworks, encompassing devices utilized for satellite communications. Significantly, in the last half-year, a multitude of occurrences have been documented across diverse regions. These instances comprise locations such as the central and eastern Mediterranean Sea, with notable proximity to the Suez Canal, Istanbul, Cyprus, and Malta, alongside the Persian Gulf near Dammam, KSA, and the coastal vicinity of Brazil. Furthermore, it is important to note that Automatic Identification Systems (AIS) operate on non-secure VHF-FM channels, utilizing open, unencrypted, and unprotected radio systems. This inherent vulnerability makes AIS signals susceptible to spoofing, which can lead to the dissemination of inaccurate or even missing AIS data[53].

In this chapter we are taking real sea route data, securing it using Order Preserving Encryption library written in Python language. Python is a versatile and widely used programming language known for its high-level nature. It is designed to prioritize code readability, utilizing the off-side rule to enforce significant indentation. Python is dynamically typed and incorporates automatic garbage collection.

To facilitate the storage of the collected data, we have employed the MySQL database. MySQL, a freely available relational database management system (RDBMS), utilizes structured query language (SQL) for efficient data administration in a database. MySQL is well-suited for applications of various sizes, accommodating both small-scale and large-scale environments.

Building upon the foundational architecture discussed in the previous chapter, this chapter delves into the practical aspects and intricacies of executing the various stages of the technique.

5.2 Implementation

A comprehensive exploration of the implementation details of each phase in the proposed technique is as follows:

5.2.1 Reference Data Collection

The collection of coordinates for the vessel's intended route is performed using a third-party API, searoutes.com.

5.2.1.1 The Searoute API

The API facilitates the retrieval of the necessary geospatial data required for the vessel's navigation plan in different languages like Ruby, PhP, Python, Node, C#, Javascript etc. The API provides various endpoints and categorized into three collections; report, search, and plan. These collections serve different purposes as follows:

1. The report collection gathers historical information, such as past vessel routes or voyages, emissions data for previous shipments, and past weather information.

2. The search collection enables users to perform forward and reverse geocoding for various locations, including ports, airports, and zip codes. Additionally, it provides information about vessels, their services, and carrier details.

3. The plan collection focuses on future-oriented functionalities. Users can utilize these APIs to plan a vessel voyage, estimate CO2 emissions for a shipment, obtain weather forecasts along a route, and create an itinerary for a shipment.

Below are the different endpoints provided by the searoute API:

a) CO2 Service: Calculates the CO2e emissions of a container on various modes of transport such as SEA, AIR, RAIL, and ROAD.

b) Vessel Service: Provides real-time vessel positions and estimated travel times to their next destination.

c) Weather Service: Allows user to access weather data points in the past, at specific locations, or along a given route.

d) Geocoding Service: Searches for points of interest, such as ports or airports, and provides their geolocations.

e) Search Service: Searches for entities used in other endpoints, including carriers or services.

f) Vessel Service: Enables user to search for vessels by name and provides relevant information about fleets.

g) CO2 Service: Provides CO2e emissions statistics for a shipment, considering the carrier and port pair. This is useful for estimating emissions when planning future shipments.

h) Itinerary Service: Searches for itineraries operated by carriers for freight transport. Can be combined with the CO2 service to explore transport routes.

i) Route Service: Allows user to plan vessel voyages, calculate distances, and obtain travel times between ports or points, whether on land or sea.

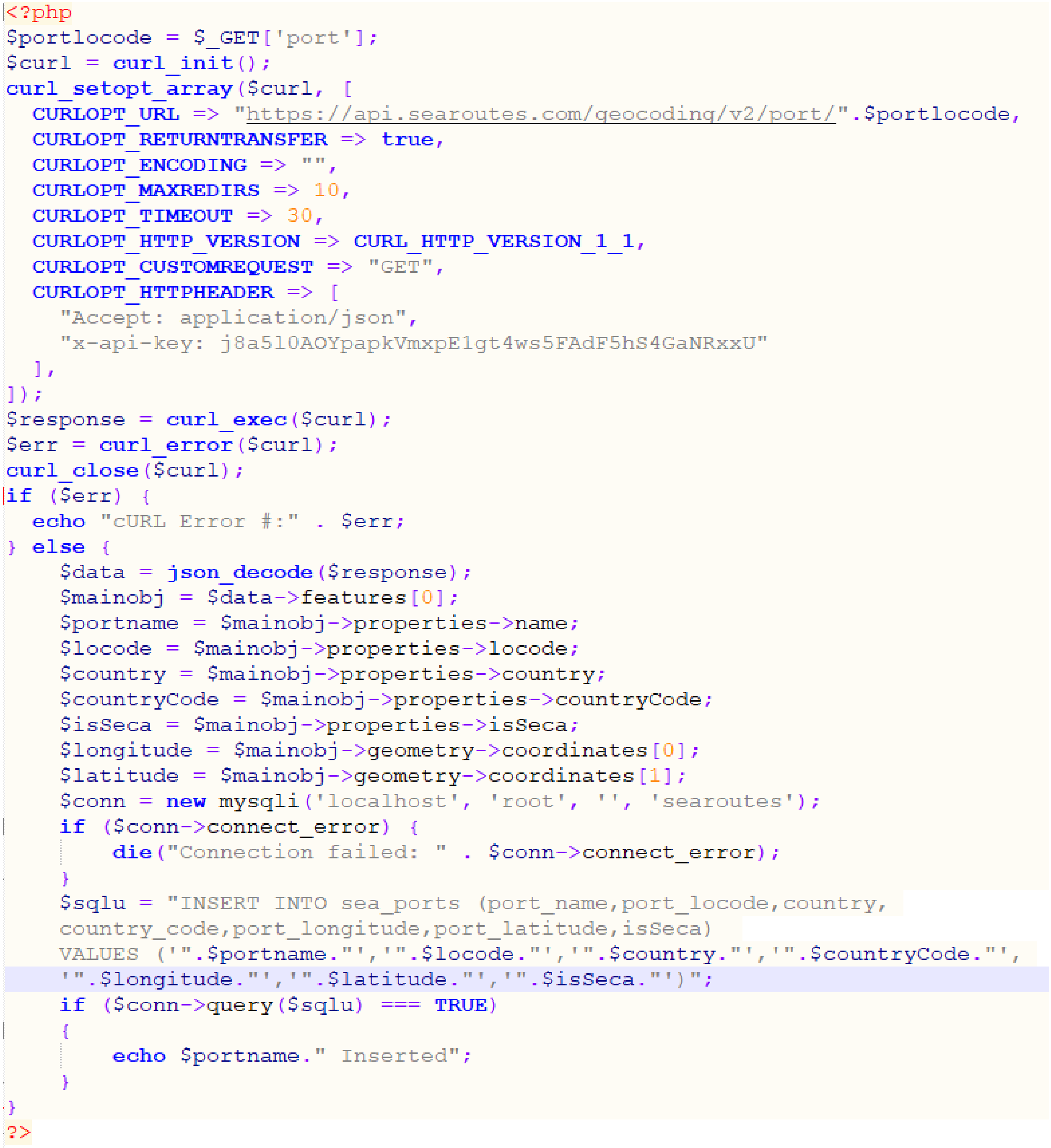
j) Vessel Service: Provides estimated travel times to the next destination for vessels.

k) Weather Service: Allows user to explore weather data points in the future, at specific locations, or along a given route.

Out of the aforementioned endpoints, we utilized the Geocoding Service and Route Service endpoints. To make API calls, it was necessary to obtain API keys from the searoute team, which were provided upon request. To store retrieved data, a database called ‘searoutes’ was created via phpMyAdmin. PhpMyAdmin is a widely used administration tool for MySQL and MariaDB databases. It is a free and open source web application, predominantly developed in PHP, that has gained significant popularity, particularly among web hosting services.

5.2.1.2 The Geocoding Service Endpoint

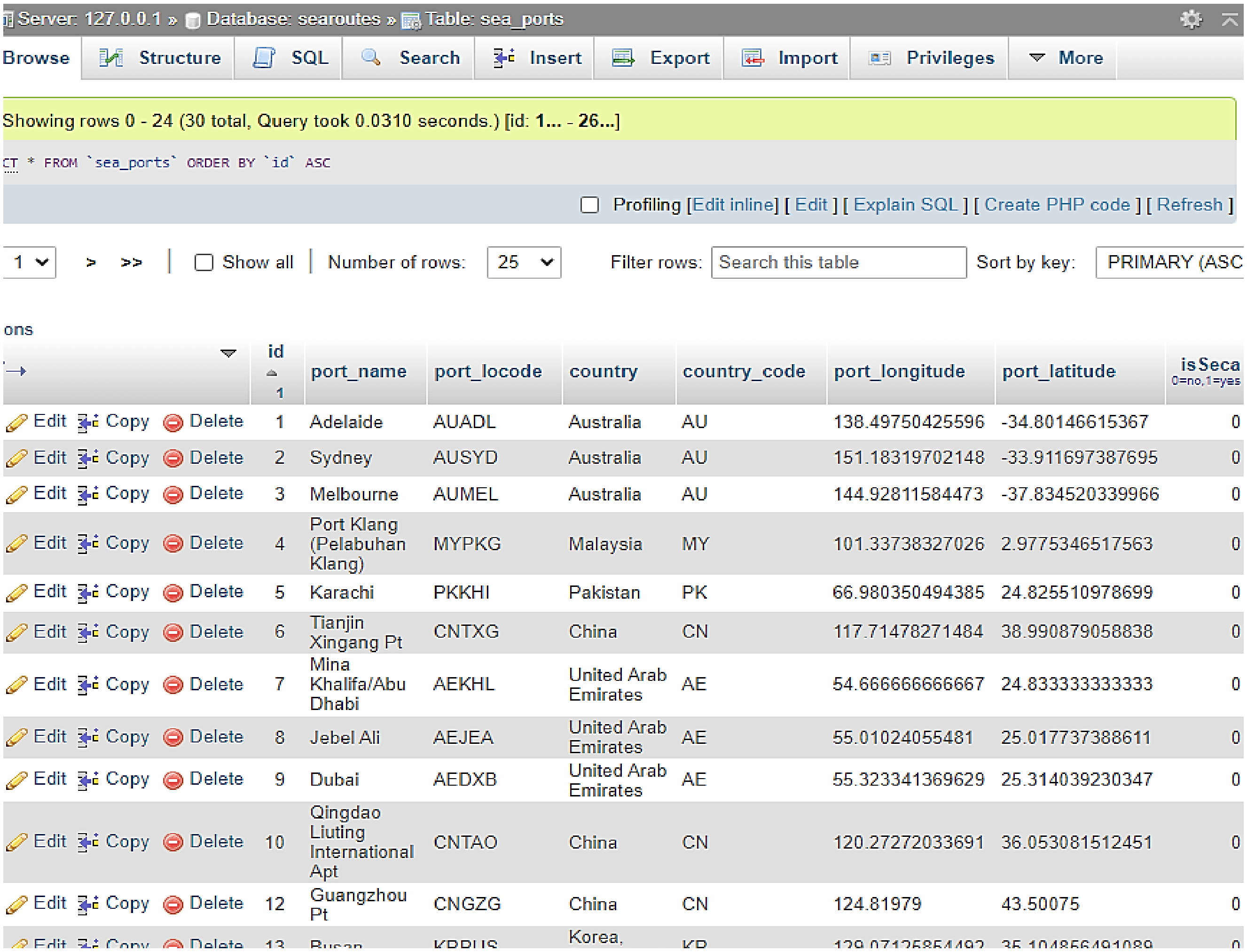
This endpoint by searoutes.com performs forward geocoding, allowing user to provide a name or locode and retrieve matching ports. The locode is passed as parameter to geocoding service endpoint and CURL is used to retrieve API response. CURL is a versatile command-line tool and application library that enables data transfer to and from a server. It utilizes URL syntax, including HTTP and HTTPS, for communication purposes. The returned information from searoute API includes the port's name, locode, country, coordinates, and whether the port is located in an ECA zone or not. We used endpoint provided in PhP language as shown in Figure 5.1.



```php
<?php
$portlocode = $_GET['port'];
$curl = curl_init();
curl_setopt_array($curl, [
  CURLOPT_URL => "https://api.searoutes.com/geocoding/v2/port/".$portlocode,
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_ENCODING => "",
  CURLOPT_MAXREDIRS => 10,
  CURLOPT_TIMEOUT => 30,
  CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
  CURLOPT_CUSTOMREQUEST => "GET",
  CURLOPT_HTTPHEADER => [
    "Accept: application/json",
    "x-api-key: j8a510AOYpapkVmxpE1gt4ws5FAdF5hS4GaNRxxU"
  ],
]);
$response = curl_exec($curl);
$err = curl_error($curl);
curl_close($curl);
if ($err) {
  echo "cURL Error #:" . $err;
} else {
    $data = json_decode($response);
    $mainobj = $data->features[0];
    $portname = $mainobj->properties->name;
    $locode = $mainobj->properties->locode;
    $country = $mainobj->properties->country;
    $countryCode = $mainobj->properties->countryCode;
    $isSeca = $mainobj->properties->isSeca;
    $longitude = $mainobj->geometry->coordinates[0];
    $latitude = $mainobj->geometry->coordinates[1];
    $conn = new mysqli('localhost', 'root', '', 'searoutes');
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    $sqlu = "INSERT INTO sea_ports (port_name,port_locode,country,
    country_code,port_longitude,port_latitude,isSeca)
    VALUES ('".$portname."','".$locode."','".$country."','".$countryCode."',
    '".$longitude."','".$latitude."','".$isSeca."')";
    if ($conn->query($sqlu) === TRUE)
    {
        echo $portname." Inserted";
    }
}
?>
```

Figure 5.1 Port Data Collection PhP Code

Data was collected for 30 different ports and inserted in ‘sea_ports’ table of ‘searoutes’ database as depicted in Figure 5.2.



Showing rows 0 - 24 (30 total, Query took 0.0310 seconds.) [id: 1... - 26...]

```
CT * FROM `sea_ports` ORDER BY `id` ASC
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

1 > >> | ☐ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: PRIMARY (ASC

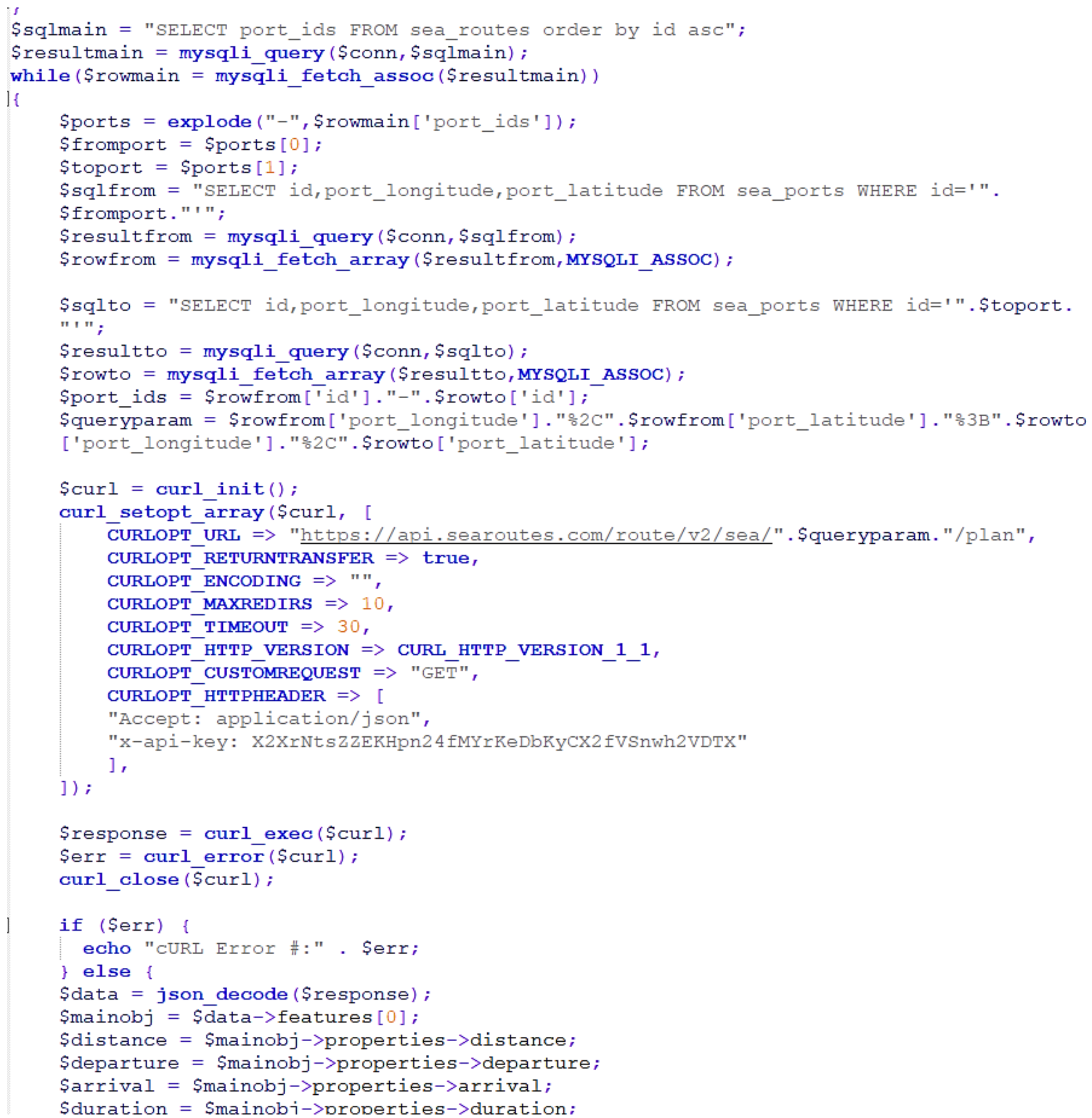
ons

	id	port_name	port_locode	country	country_code	port_longitude	port_latitude	isSeca 0=no,1=yes
Edit Copy Delete	1	Adelaide	AUADL	Australia	AU	138.49750425596	-34.80146615367	0
Edit Copy Delete	2	Sydney	AUSYD	Australia	AU	151.18319702148	-33.911697387695	0
Edit Copy Delete	3	Melbourne	AUMEL	Australia	AU	144.92811584473	-37.834520339966	0
Edit Copy Delete	4	Port Klang (Pelabuhan Klang)	MYPKG	Malaysia	MY	101.33738327026	2.9775346517563	0
Edit Copy Delete	5	Karachi	PKKHI	Pakistan	PK	66.980350494385	24.825510978699	0
Edit Copy Delete	6	Tianjin Xingang Pt	CNTXG	China	CN	117.71478271484	38.990879058838	0
Edit Copy Delete	7	Mina Khalifa/Abu Dhabi	AEKHL	United Arab Emirates	AE	54.666666666667	24.833333333333	0
Edit Copy Delete	8	Jebel Ali	AEJEA	United Arab Emirates	AE	55.01024055481	25.017737388611	0
Edit Copy Delete	9	Dubai	AEDXB	United Arab Emirates	AE	55.323341369629	25.314039230347	0
Edit Copy Delete	10	Qingdao Liuting International Apt	CNTAO	China	CN	120.27272033691	36.053081512451	0
Edit Copy Delete	12	Guangzhou Pt	CNGZG	China	CN	124.81979	43.50075	0
Edit Copy Delete	13	Busan	KRPUS	Korea,	KR	129.07125854492	35.104856491089	0

Figure 5.2 Port Data Collection MySQL Table

5.2.1.3 The Route Service Endpoint

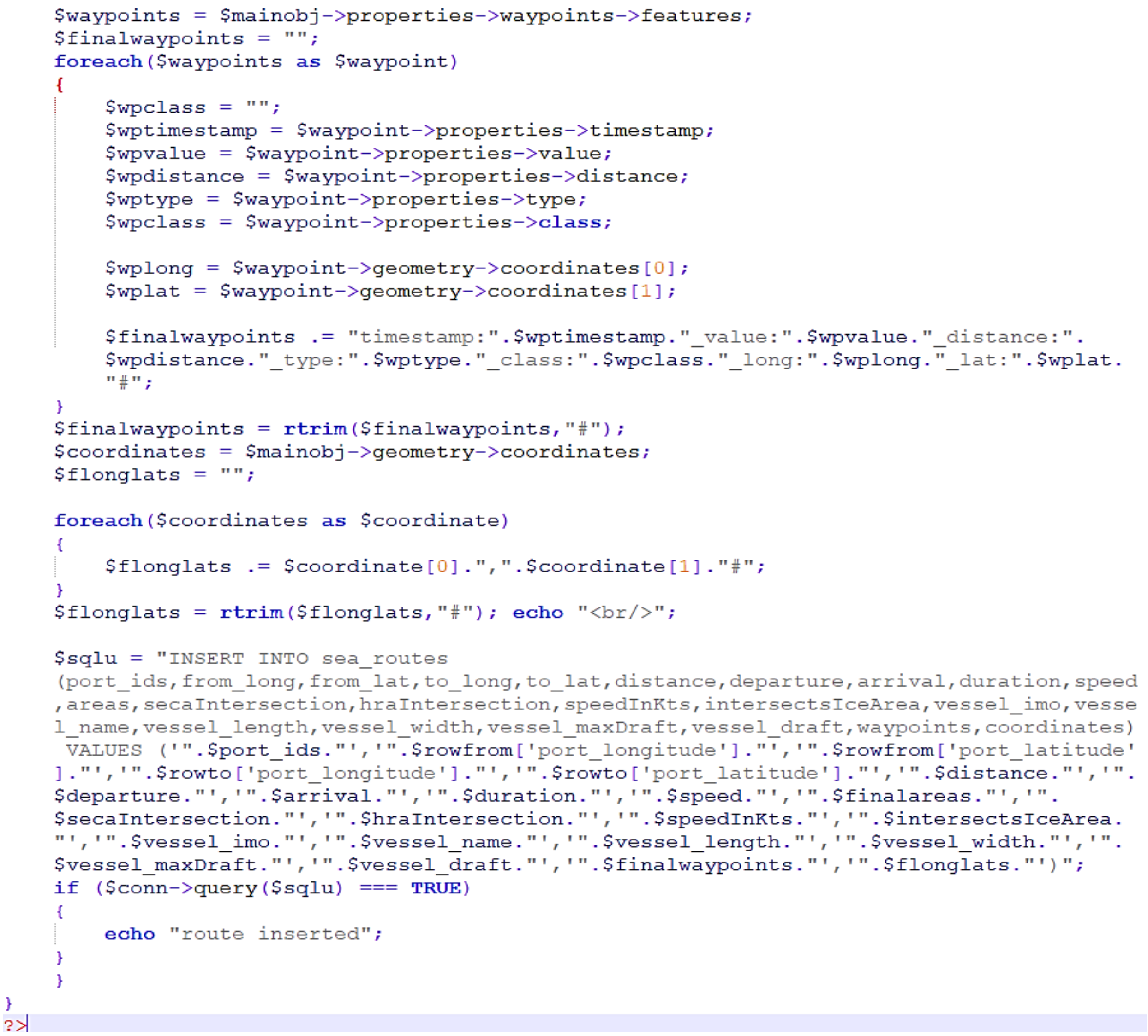
By leveraging the provided endpoint, we can obtain the sea route between a specified source location and target location using their respective locode information. It includes details such as duration (in milliseconds), the route distance (in meters), and information about the zones traversed. The returned route is the shortest path taking into account port entries and traffic separation schemes. We utilized the PHP version of endpoint and customized it to align with our specific requirements.



```
;
$sqlmain = "SELECT port_ids FROM sea_routes order by id asc";
$resultmain = mysqli_query($conn,$sqlmain);
while($rowmain = mysqli_fetch_assoc($resultmain))
{
    $ports = explode("-",$rowmain['port_ids']);
    $fromport = $ports[0];
    $toport = $ports[1];
    $sqlfrom = "SELECT id,port_longitude,port_latitude FROM sea_ports WHERE id='".
    $fromport."'";
    $resultfrom = mysqli_query($conn,$sqlfrom);
    $rowfrom = mysqli_fetch_array($resultfrom,MYSQLI_ASSOC);

    $sqlto = "SELECT id,port_longitude,port_latitude FROM sea_ports WHERE id='".$toport.
    "'";
    $resultto = mysqli_query($conn,$sqlto);
    $rowto = mysqli_fetch_array($resultto,MYSQLI_ASSOC);
    $port_ids = $rowfrom['id']."-".$rowto['id'];
    $queryparam = $rowfrom['port_longitude']."%2C".$rowfrom['port_latitude']."%3B".$rowto
    ['port_longitude']."%2C".$rowto['port_latitude'];

    $curl = curl_init();
    curl_setopt_array($curl, [
        CURLOPT_URL => "https://api.searoutes.com/route/v2/sea/".$queryparam."/plan",
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_ENCODING => "",
        CURLOPT_MAXREDIRS => 10,
        CURLOPT_TIMEOUT => 30,
        CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
        CURLOPT_CUSTOMREQUEST => "GET",
        CURLOPT_HTTPHEADER => [
        "Accept: application/json",
        "x-api-key: X2XrNtsZZEKHpn24fMYrKeDbKyCX2fVSnwh2VDTX"
        ],
    ]);

    $response = curl_exec($curl);
    $err = curl_error($curl);
    curl_close($curl);

    if ($err) {
      echo "cURL Error #:" . $err;
    } else {
    $data = json_decode($response);
    $mainobj = $data->features[0];
    $distance = $mainobj->properties->distance;
    $departure = $mainobj->properties->departure;
    $arrival = $mainobj->properties->arrival;
    $duration = $mainobj->properties->duration;
```

Figure 5.3 Sea Route Collection PhP Code

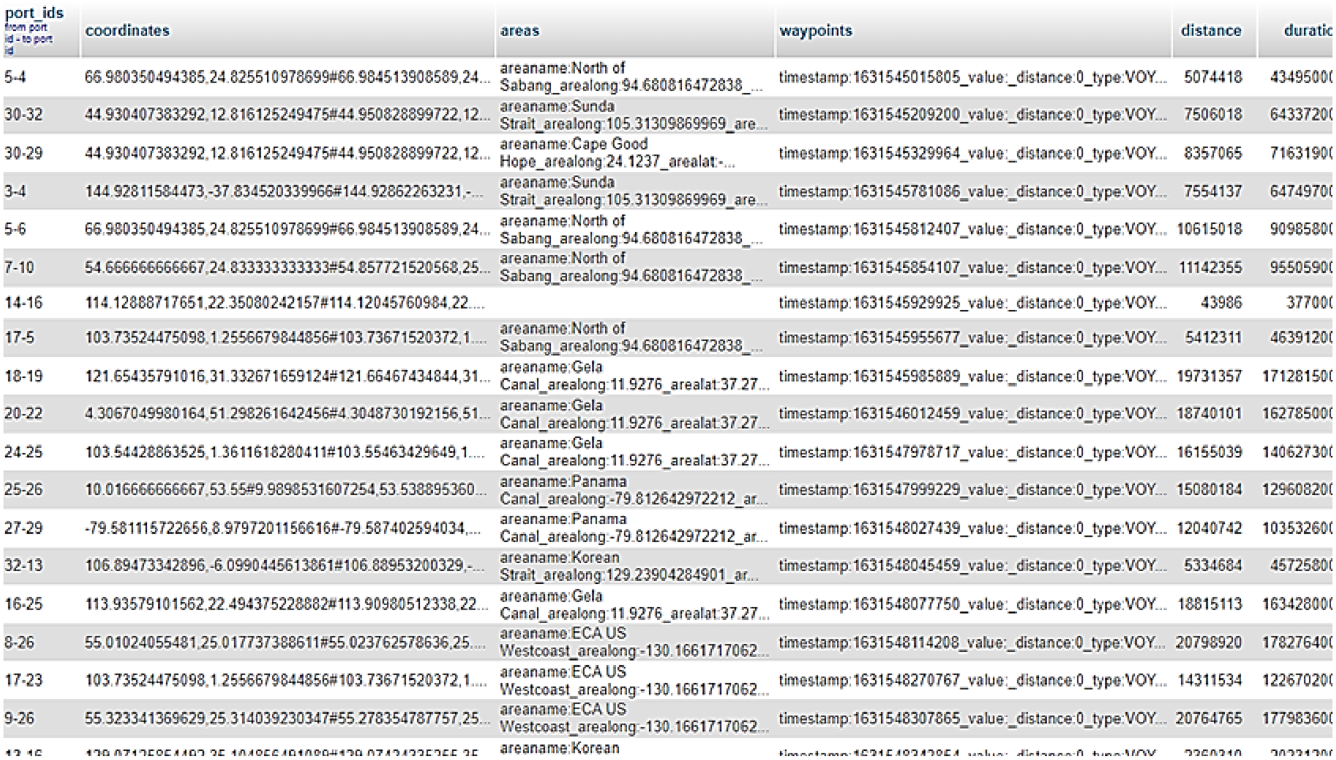
In Figure 5.3, the port IDs are utilized to retrieve port locodes from the "sea_ports" table. The source and destination locodes are then provided as parameters to the 'plan' searoute endpoint. The results are obtained using CURL, and the response from CURL is parsed and assigned to the corresponding variables.



```
    $waypoints = $mainobj->properties->waypoints->features;
    $finalwaypoints = "";
    foreach($waypoints as $waypoint)
    {
        $wpclass = "";
        $wptimestamp = $waypoint->properties->timestamp;
        $wpvalue = $waypoint->properties->value;
        $wpdistance = $waypoint->properties->distance;
        $wptype = $waypoint->properties->type;
        $wpclass = $waypoint->properties->class;

        $wplong = $waypoint->geometry->coordinates[0];
        $wplat = $waypoint->geometry->coordinates[1];

        $finalwaypoints .= "timestamp:".$wptimestamp."_value:".$wpvalue."_distance:".
        $wpdistance."_type:".$wptype."_class:".$wpclass."_long:".$wplong."_lat:".$wplat.
        "#";
    }
    $finalwaypoints = rtrim($finalwaypoints,"#");
    $coordinates = $mainobj->geometry->coordinates;
    $flonglats = "";

    foreach($coordinates as $coordinate)
    {
        $flonglats .= $coordinate[0].",".$coordinate[1]."#";
    }
    $flonglats = rtrim($flonglats,"#"); echo "<br/>";

    $sqlu = "INSERT INTO sea_routes
    (port_ids,from_long,from_lat,to_long,to_lat,distance,departure,arrival,duration,speed
    ,areas,secaIntersection,hraIntersection,speedInKts,intersectsIceArea,vessel_imo,vesse
    l_name,vessel_length,vessel_width,vessel_maxDraft,vessel_draft,waypoints,coordinates)
     VALUES ('".$port_ids."','".$rowfrom['port_longitude']."','".$rowfrom['port_latitude'
    ]."','".$rowto['port_longitude']."','".$rowto['port_latitude']."','".$distance."','".
    $departure."','".$arrival."','".$duration."','".$speed."','".$finalareas."','".
    $secaIntersection."','".$hraIntersection."','".$speedInKts."','".$intersectsIceArea.
    "','".$vessel_imo."','".$vessel_name."','".$vessel_length."','".$vessel_width."','".
    $vessel_maxDraft."','".$vessel_draft."','".$finalwaypoints."','".$flonglats."')";
    if ($conn->query($sqlu) === TRUE)
    {
        echo "route inserted";
    }
    }
}
?>
```

Figure 5.4 Sea Route Insertion PhP Code

Figure 5.4 demonstrates the process through which we successfully manipulated and organized the raw data obtained from the API to align it with the structure of our database columns. A total of 30 sea routes were surveyed and the corresponding data was collected, which was inserted into the 'sea_routes' table within the ‘searoutes’ database as shown in Figure 5.5.



port_ids from port id - to port id	coordinates	areas	waypoints	distance	duratio
5-4	66.980350494385,24.825510978699#66.984513908589,24...	areaname:North of Sabang_arealong:94.680816472838_...	timestamp:1631545015805_value:_distance:0_type:VOY...	5074418	4349500
30-32	44.930407383292,12.816125249475#44.950828899722,12...	areaname:Sunda Strait_arealong:105.31309869969_are...	timestamp:1631545209200_value:_distance:0_type:VOY...	7506018	6433720
30-29	44.930407383292,12.816125249475#44.950828899722,12...	areaname:Cape Good Hope_arealong:24.1237_arealat:-...	timestamp:1631545329964_value:_distance:0_type:VOY...	8357065	7163190
3-4	144.92811584473,-37.834520339966#144.92862263231,-...	areaname:Sunda Strait_arealong:105.31309869969_are...	timestamp:1631545781086_value:_distance:0_type:VOY...	7554137	6474970
5-6	66.980350494385,24.825510978699#66.984513908589,24...	areaname:North of Sabang_arealong:94.680816472838_...	timestamp:1631545812407_value:_distance:0_type:VOY...	10615018	9098580
7-10	54.666666666667,24.833333333333#54.857721520568,25...	areaname:North of Sabang_arealong:94.680816472838_...	timestamp:1631545854107_value:_distance:0_type:VOY...	11142355	9550590
14-16	114.12888717651,22.35080242157#114.12045760984,22....		timestamp:1631545929925_value:_distance:0_type:VOY...	43986	37700
17-5	103.73524475098,1.2556679844856#103.73671520372,1....	areaname:North of Sabang_arealong:94.680816472838_...	timestamp:1631545955677_value:_distance:0_type:VOY...	5412311	4639120
18-19	121.65435791016,31.332671659124#121.66467434844,31...	areaname:Gela Canal_arealong:11.9276_arealat:37.27...	timestamp:1631545985889_value:_distance:0_type:VOY...	19731357	17128150
20-22	4.3067049980164,51.298261642456#4.3048730192156,51...	areaname:Gela Canal_arealong:11.9276_arealat:37.27...	timestamp:1631546012459_value:_distance:0_type:VOY...	18740101	16278500
24-25	103.54428863525,1.3611618280411#103.55463429649,1....	areaname:Gela Canal_arealong:11.9276_arealat:37.27...	timestamp:1631547978717_value:_distance:0_type:VOY...	16155039	14062730
25-26	10.016666666667,53.55#9.9898531607254,53.538895360...	areaname:Panama Canal_arealong:-79.812642972212_ar...	timestamp:1631547999229_value:_distance:0_type:VOY...	15080184	12960820
27-29	-79.581115722656,8.9797201156616#-79.587402594034,...	areaname:Panama Canal_arealong:-79.812642972212_ar...	timestamp:1631548027439_value:_distance:0_type:VOY...	12040742	10353260
32-13	106.89473342896,-6.0990445613861#106.88953200329,-...	areaname:Korean Strait_arealong:129.23904284901_ar...	timestamp:1631548045459_value:_distance:0_type:VOY...	5334684	4572580
16-25	113.93579101562,22.494375228882#113.90980512338,22...	areaname:Gela Canal_arealong:11.9276_arealat:37.27...	timestamp:1631548077750_value:_distance:0_type:VOY...	18815113	16342800
8-26	55.01024055481,25.017737388611#55.023762578636,25....	areaname:ECA US Westcoast_arealong:-130.1661717062...	timestamp:1631548114208_value:_distance:0_type:VOY...	20798920	17827640
17-23	103.73524475098,1.2556679844856#103.73671520372,1....	areaname:ECA US Westcoast_arealong:-130.1661717062...	timestamp:1631548270767_value:_distance:0_type:VOY...	14311534	12267020
9-26	55.323341369629,25.314039230347#55.278354787757,25...	areaname:ECA US Westcoast_arealong:-130.1661717062...	timestamp:1631548307865_value:_distance:0_type:VOY...	20764765	17798360
[illegible]	[illegible]	areaname:Korean	[illegible]	[illegible]	[illegible]

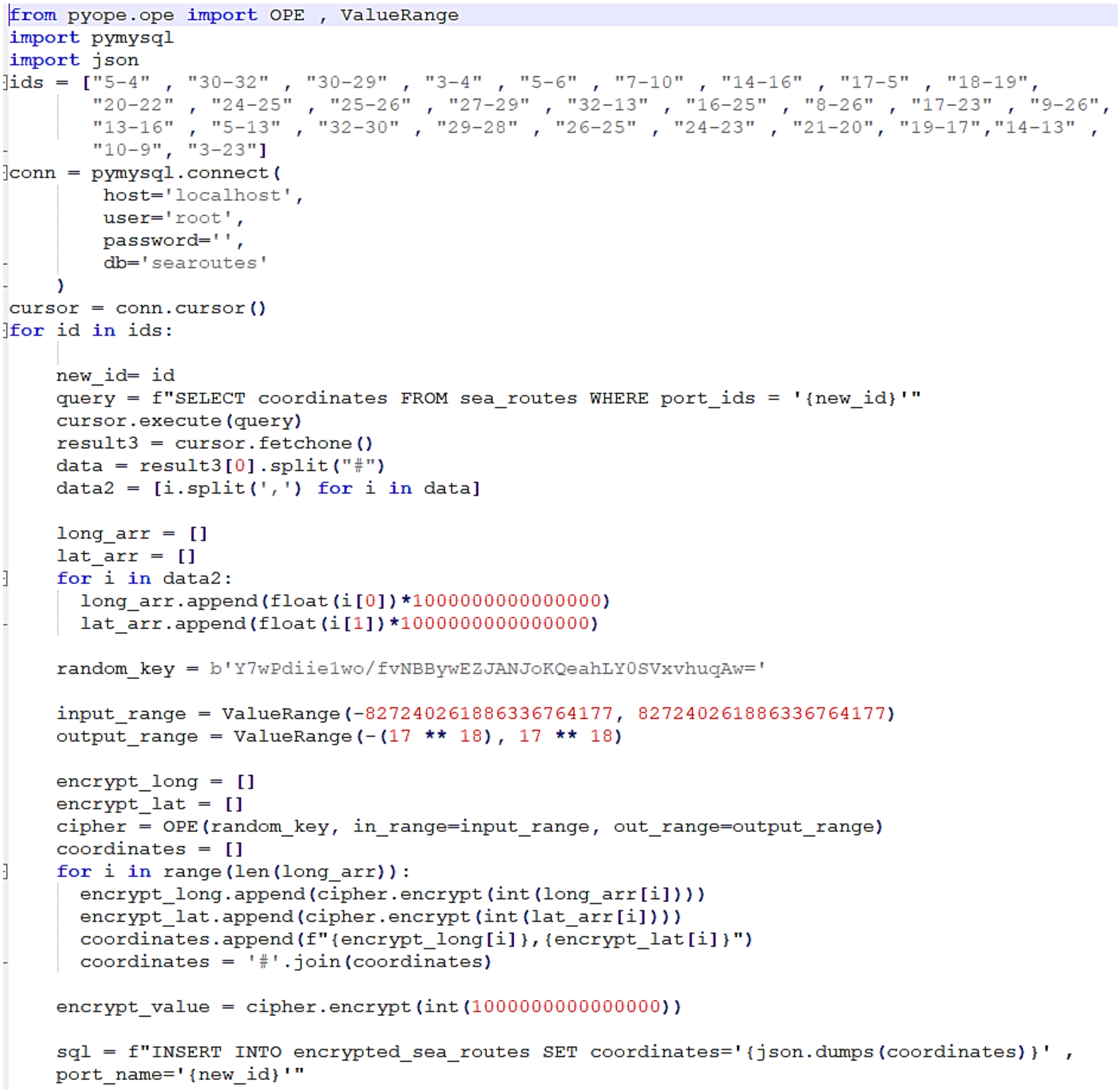
Figure 5.5 Sea Routes Data Collection MySQL Table

5.2.2 Encryption

The sea route data collected in the previous step has been encrypted using the encryption technique offered by the OPE library in Python. The code, as presented in Figure 5.6, retrieves the sea route data from the database and applies encryption using OPE, a symmetric encryption technique that allows for sorting of encrypted data and enables range queries while preserving the order. The encryption algorithm utilizes a private key, input range, and output range to create a cipher object, and encryption is performed by invoking the 'encrypt' function on the cipher object.

In the default OPE technique, a randomly generated private key is used. However, to fulfill the availability design goal and enable decryption, we are using a fixed private key. Additionally, the default input and output ranges are set as (0 to 2^15-1) and (0 to 2^31-1) respectively, which are suitable for encrypting small amounts of data consisting of 15 bits. However, since we are encrypting coordinates that are at least 45 bits in size, we had to increase the input and output ranges, which are referred to as the Plaintext space and Cyphertext space in Boldyreva's scheme.

One limitation of the library is that it only supports integer values, while sea route coordinates are of float type. To address this limitation, we implemented a solution by converting the floating-point coordinates to integer data type. Additionally, we took into account the varying lengths of latitudes and longitudes to ensure accurate encryption and decryption of the coordinates.

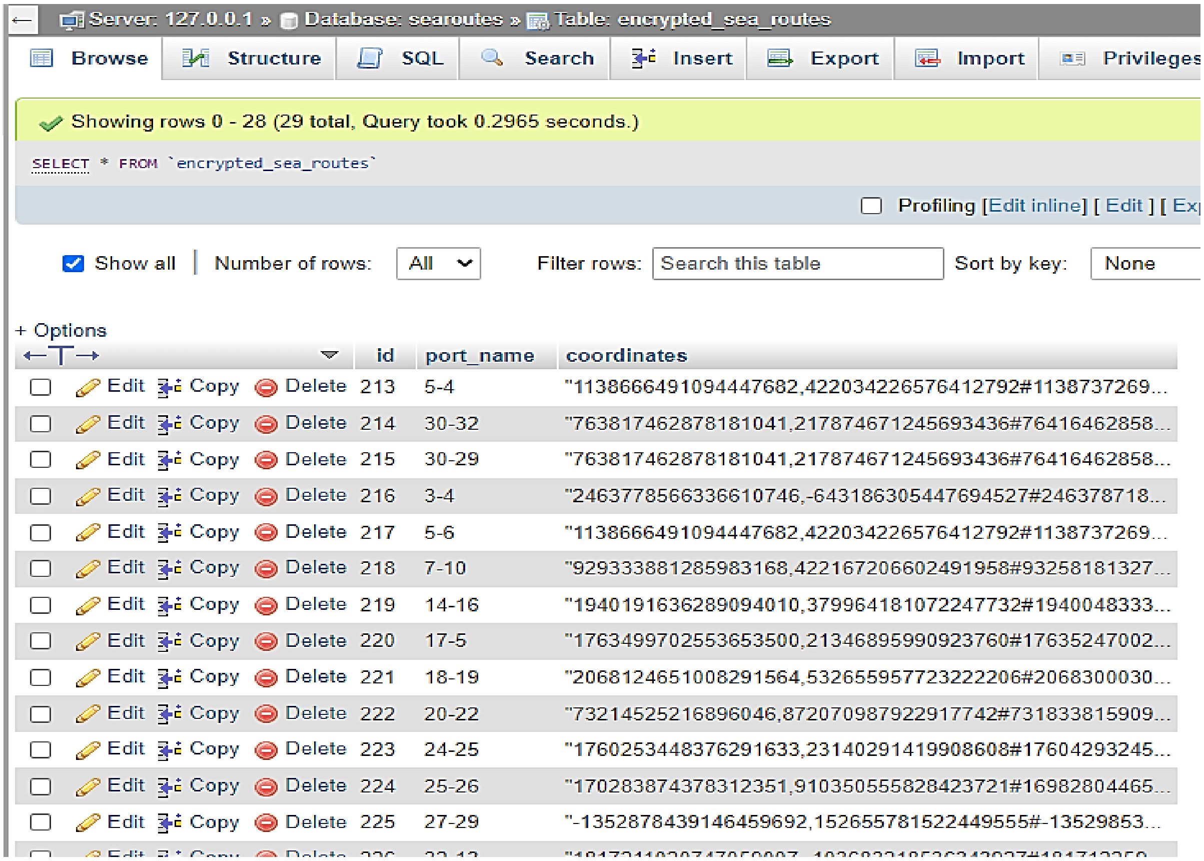


```python
from pyope.ope import OPE , ValueRange
import pymysql
import json
ids = ["5-4" , "30-32" , "30-29" , "3-4" , "5-6" , "7-10" , "14-16" , "17-5" , "18-19",
       "20-22" , "24-25" , "25-26" , "27-29" , "32-13" , "16-25" , "8-26" , "17-23" , "9-26",
       "13-16" , "5-13" , "32-30" , "29-28" , "26-25" , "24-23" , "21-20", "19-17","14-13" ,
       "10-9", "3-23"]
conn = pymysql.connect(
        host='localhost',
        user='root',
        password='',
        db='searoutes'
    )
cursor = conn.cursor()
for id in ids:

    new_id= id
    query = f"SELECT coordinates FROM sea_routes WHERE port_ids = '{new_id}'"
    cursor.execute(query)
    result3 = cursor.fetchone()
    data = result3[0].split("#")
    data2 = [i.split(',') for i in data]

    long_arr = []
    lat_arr = []
    for i in data2:
      long_arr.append(float(i[0])*1000000000000000)
      lat_arr.append(float(i[1])*1000000000000000)

    random_key = b'Y7wPdiie1wo/fvNBBywEZJANJoKQeahLY0SVxvhuqAw='

    input_range = ValueRange(-827240261886336764177, 827240261886336764177)
    output_range = ValueRange(-(17 ** 18), 17 ** 18)

    encrypt_long = []
    encrypt_lat = []
    cipher = OPE(random_key, in_range=input_range, out_range=output_range)
    coordinates = []
    for i in range(len(long_arr)):
      encrypt_long.append(cipher.encrypt(int(long_arr[i])))
      encrypt_lat.append(cipher.encrypt(int(lat_arr[i])))
      coordinates.append(f"{encrypt_long[i]},{encrypt_lat[i]}")
      coordinates = '#'.join(coordinates)

    encrypt_value = cipher.encrypt(int(1000000000000000))

    sql = f"INSERT INTO encrypted_sea_routes SET coordinates='{json.dumps(coordinates)}' ,
    port_name='{new_id}'"
```

Figure 5.6 Data Encryption using Python OPE

5.2.3 Reference Data Storage

The latitude-longitude pairs for each sea route are encrypted using the chosen encryption technique, and the resulting encrypted values are inserted into the database, as indicated in Figure 5.7. This step ensures the confidentiality and security of the geographical data, as the encrypted values cannot be directly interpreted or accessed by unauthorized users. By securely storing the encrypted latitude-longitude pairs in the database, we maintain the integrity and privacy of the sea route information, while allowing authorized users to retrieve and process the data as needed for further analysis and navigation purposes.


←
Server: 127.0.0.1 » Database: searoutes » Table: encrypted_sea_routes
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges

✓ Showing rows 0 - 28 (29 total, Query took 0.2965 seconds.)

```
SELECT * FROM `encrypted_sea_routes`
```

□ Profiling [Edit inline] [Edit] [Ex

☑ Show all | Number of rows: All

Filter rows: Search this table

Sort by key: None

		id	port_name	coordinates
□	Edit Copy Delete	213	5-4	"1138666491094447682,422034226576412792#1138737269...
□	Edit Copy Delete	214	30-32	"763817462878181041,217874671245693436#76416462858...
□	Edit Copy Delete	215	30-29	"763817462878181041,217874671245693436#76416462858...
□	Edit Copy Delete	216	3-4	"2463778566336610746,-643186305447694527#246378718...
□	Edit Copy Delete	217	5-6	"1138666491094447682,422034226576412792#1138737269...
□	Edit Copy Delete	218	7-10	"929333881285983168,422167206602491958#93258181327...
□	Edit Copy Delete	219	14-16	"1940191636289094010,379964181072247732#1940048333...
□	Edit Copy Delete	220	17-5	"1763499702553653500,213468959909237760#17635247002...
□	Edit Copy Delete	221	18-19	"2068124651008291564,532655957723222206#2068300030...
□	Edit Copy Delete	222	20-22	"73214525216896046,872070987922917742#731833815909...
□	Edit Copy Delete	223	24-25	"1760253448376291633,231402914199086608#17604293245...
□	Edit Copy Delete	224	25-26	"170283874378312351,910350555828423721#16982804465...
□	Edit Copy Delete	225	27-29	"-1352878439146459692,152655781522449555#-13529853...

Figure 5.7 Encrypted Data Storage

5.2.4 User Command and Interface

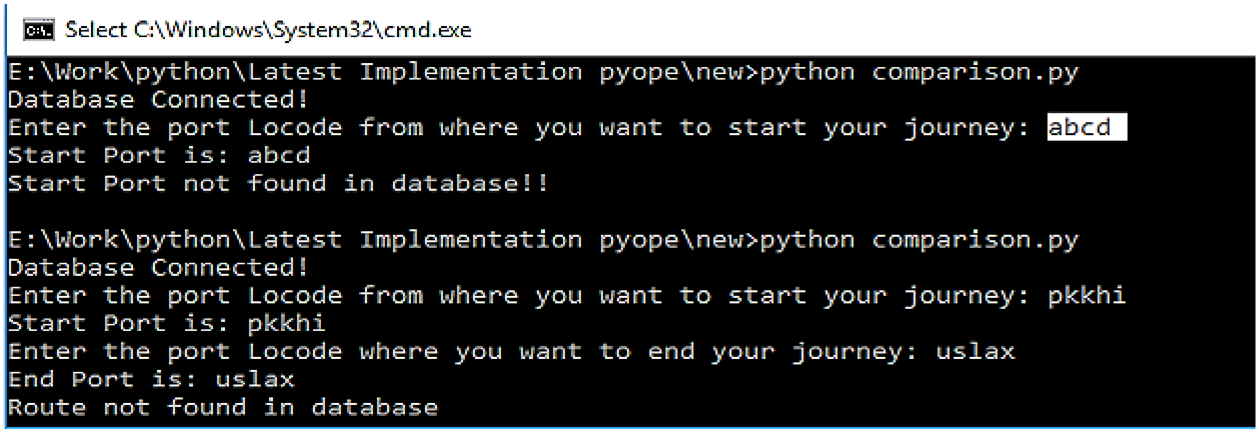
The user is presented with a user-friendly graphical user interface (UI) that allows them to interact with the system. Within the UI, the user is prompted to enter the locodes of the source and destination ports, as well as the batch of data they wish to perform a comparison for. This user input is crucial for the system to accurately retrieve the relevant sea route data and conduct the necessary comparisons.

The implemented code first establishes a connection with the database to ensure seamless data retrieval and processing. This connection serves as a vital link between the system and the stored sea route information. Once the connection is successfully established, the code proceeds to validate the user's input.

During the validation process, the code checks the accuracy and appropriateness of the entered locodes. It verifies whether the locodes correspond to valid source and destination ports within the database. Additionally, the code ensures that the specified sea routes are present in the database, as this information is crucial for conducting the desired comparison.

In the event that the user enters an incorrect or invalid locode, or if the specified routes are not found in the database, the system promptly displays an error message, as demonstrated in Figure 5.8. This error message serves as a notification to the user, allowing them to rectify their input and ensure the accuracy of the comparison process.

By implementing these validation checks and providing informative error messages, the system aims to enhance user experience and ensure the reliability and accuracy of the subsequent comparison operations.



```
E:\Work\python\Latest Implementation pyope\new>python comparison.py
Database Connected!
Enter the port Locode from where you want to start your journey: abcd
Start Port is: abcd
Start Port not found in database!!

E:\Work\python\Latest Implementation pyope\new>python comparison.py
Database Connected!
Enter the port Locode from where you want to start your journey: pkkhi
Start Port is: pkkhi
Enter the port Locode where you want to end your journey: uslax
End Port is: uslax
Route not found in database
```

Figure 5.8 User Command and Interface

5.2.5 Data Manipulation and Encryption

Since the scope of this research does not include the hardware implementation aspect, real-time GPS data collection was not feasible. Instead, we gathered three distinct sea route data sets, each obtained at different time intervals. The purpose of collecting these datasets was to conduct an extensive comparison and verify the legitimacy of the route being followed by the vessel.

5.2.5.1 Data Set Collection and Storage

To obtain these datasets, we utilized the same searoutes.com API that was used to collect the reference data. The collected data was then inserted into three separate tables within the 'searoute' database. These tables were specifically named as 'live_sea_routes', 'new_sea_routes', and 'latest_sea_routes', representing the different time intervals at which the data was obtained.

By gathering multiple datasets over varying time periods, we aimed to analyze any discrepancies or changes in the sea routes. This approach allowed us to assess the consistency and accuracy of the vessel's route over time. To facilitate the comparison process, the code retrieves the batch of data specified by the user from all three datasets: 'live_sea_routes', 'new_sea_routes', and 'latest_sea_routes'. This ensures that we have a comprehensive representation of the sea routes across different time intervals.

5.2.5.2 Data Manipulation

Once the data batch is retrieved, an additional step is taken to add a fixed distance of 500 meters to each coordinate pair. To account for potential small deviations from the actual path caused by various factors such as weather conditions, ocean currents, or navigational errors, an offset of 500 meters is added to each coordinate pair in the retrieved data batch. This offset serves multiple purposes within the comparison process.

Firstly, it helps to mitigate the possibility of false positives during the comparison. By introducing a small offset, we create a buffer that allows for minor variations in the vessel's position without triggering a false indication of deviation. This approach acknowledges that vessels may not always follow the exact path due to external influences, but still aims to identify significant deviations from the expected route.

Secondly, the 500-meter offset provides a practical allowance for natural variations in the vessel's trajectory. It acknowledges that sea routes are not always rigidly defined lines but can have some inherent flexibility. By considering a slightly wider range around the expected path, we accommodate realistic navigational fluctuations without compromising the accuracy of the comparison.

5.2.5.3 Offset Justification

The decision to add an offset of exactly 500 meters, rather than a value above or below 500 meters, is based on the specific requirements and considerations of the comparison process.

By choosing a fixed offset of 500 meters, the intention is to introduce a noticeable change to the coordinate pairs while still maintaining a reasonable proximity to the original data. This offset value was determined through experimentation and analysis to strike a balance between sensitivity to deviations and avoiding excessive displacement of the coordinate pairs.

Adding an offset greater than 500 meters would result in a more significant shift in the coordinate pairs, potentially moving them too far away from their original positions. This could lead to a higher likelihood of false positives or inaccurate comparisons.

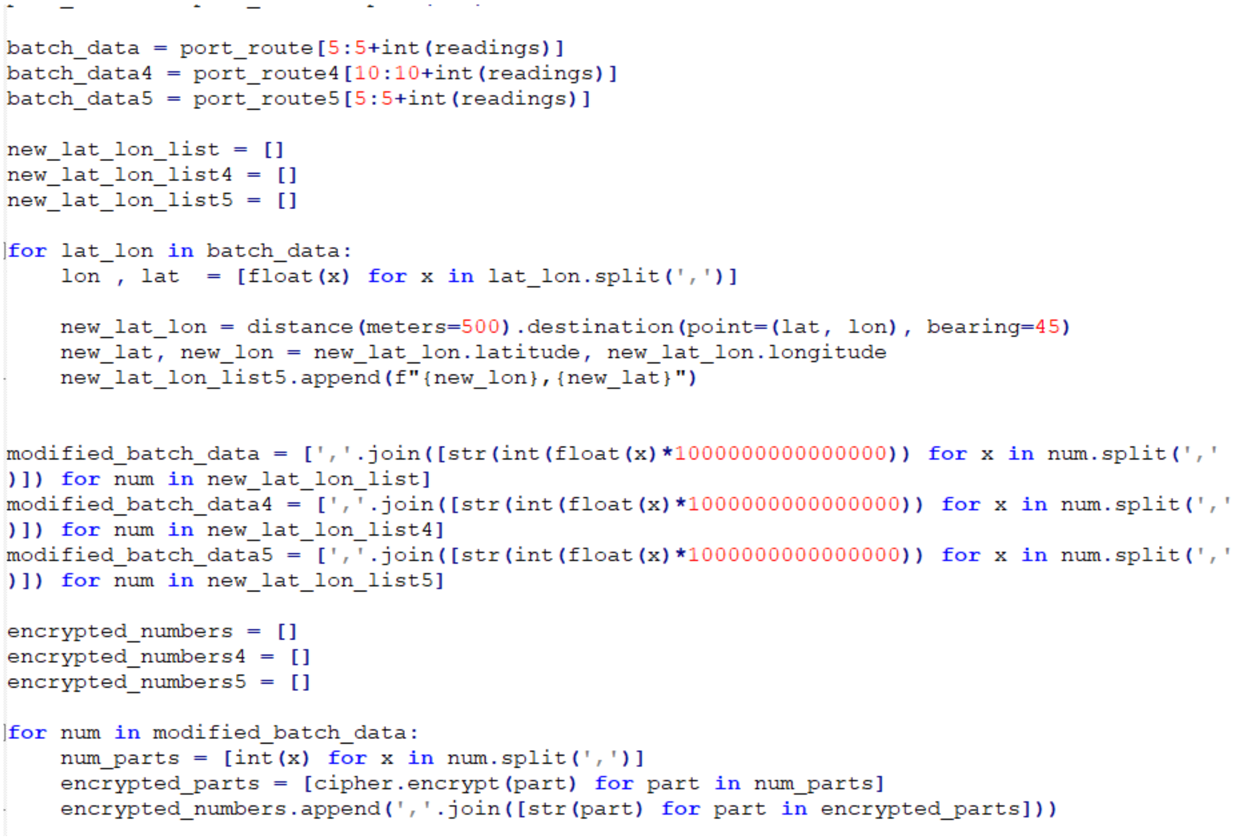
On the other hand, adding an offset below 500 meters would result in a smaller displacement that might not be sufficiently distinguishable from the original data. This could reduce the effectiveness of the comparison process in detecting meaningful deviations from the expected route.

By selecting an offset of precisely 500 meters, the coordinate pairs are shifted enough to account for minor deviations while still retaining their relationship to the original sea route.

5.2.5.4 Encryption of Modified Data

After adding the 500-meter adjustment to the retrieved batch of data, the next step involves encrypting the modified coordinates. The encryption operation follows the established OPE technique, utilizing the previously mentioned fixed private key.

By retrieving the relevant data batches, incorporating the 500-meter adjustment, and performing encryption, we establish a standardized and consistent basis for the subsequent comparison phase. Figure 5.9 visually illustrates the procedure being discussed. This preparation step sets the foundation for accurately assessing the vessel's adherence to the expected sea routes and identifying any deviations or anomalies that may arise.



```
batch_data = port_route[5:5+int(readings)]
batch_data4 = port_route4[10:10+int(readings)]
batch_data5 = port_route5[5:5+int(readings)]

new_lat_lon_list = []
new_lat_lon_list4 = []
new_lat_lon_list5 = []

for lat_lon in batch_data:
    lon , lat  = [float(x) for x in lat_lon.split(',')]

    new_lat_lon = distance(meters=500).destination(point=(lat, lon), bearing=45)
    new_lat, new_lon = new_lat_lon.latitude, new_lat_lon.longitude
    new_lat_lon_list5.append(f"{new_lon},{new_lat}")


modified_batch_data = [','.join([str(int(float(x)*1000000000000000)) for x in num.split(','
)]) for num in new_lat_lon_list]
modified_batch_data4 = [','.join([str(int(float(x)*1000000000000000)) for x in num.split(','
)]) for num in new_lat_lon_list4]
modified_batch_data5 = [','.join([str(int(float(x)*1000000000000000)) for x in num.split(','
)]) for num in new_lat_lon_list5]

encrypted_numbers = []
encrypted_numbers4 = []
encrypted_numbers5 = []

for num in modified_batch_data:
    num_parts = [int(x) for x in num.split(',')]
    encrypted_parts = [cipher.encrypt(part) for part in num_parts]
    encrypted_numbers.append(','.join([str(part) for part in encrypted_parts]))
```

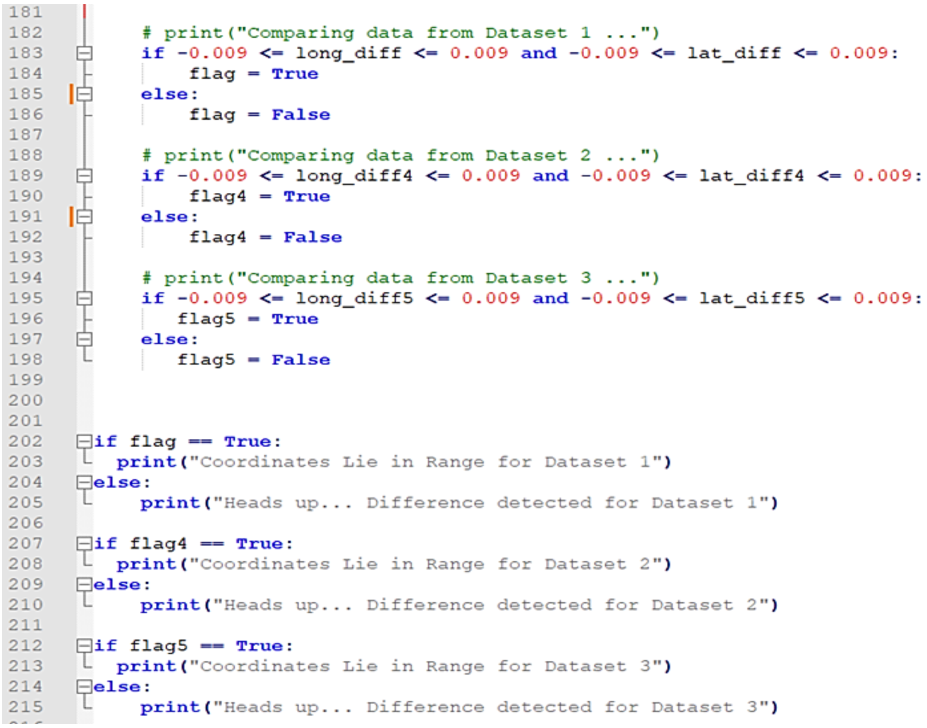
Figure 5.9 Data Manipulation and Encryption

5.2.6 Comparison

The code proceeds by retrieving the reference data for the user-specified batch from each of the three datasets. It then performs a comparison by calculating the differences between the latitude and longitude values of the retrieved data and the user's batch data.

For each comparison, the code checks if the difference in latitude and longitude values is within the buffer range. If the differences are within the range, it indicates that the vessel is following the expected route or a similar path. In such cases, a message or status indicating compliance or adherence to the route is displayed on the user's screen.

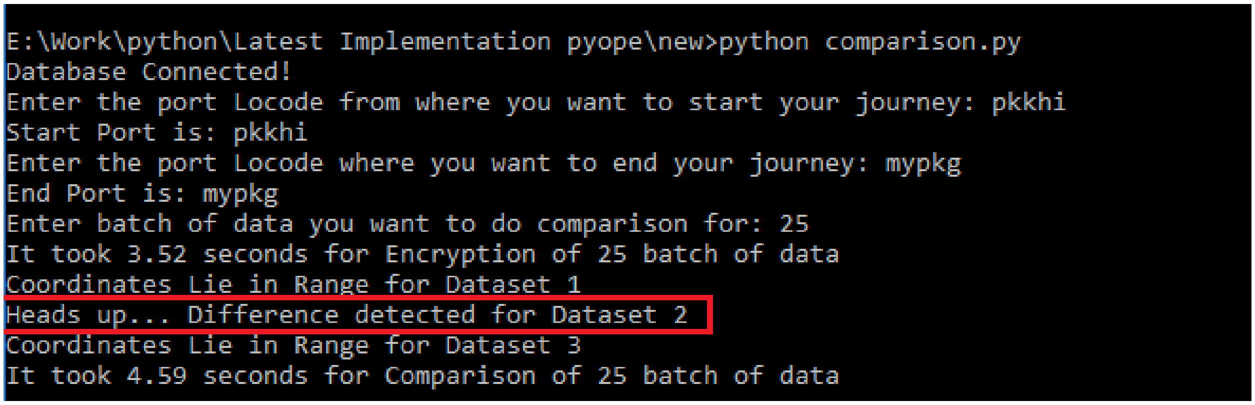
On the other hand, if the differences in latitude and longitude values exceed the buffer range, it suggests that the vessel has deviated significantly from the expected route. In such instances, a message or warning is displayed on the user's screen, alerting them to the deviation and prompting them to take necessary actions. The code snippet displayed in Figure 5.10 demonstrates the implementation of the mentioned functionality.



```
        # print("Comparing data from Dataset 1 ...")
        if -0.009 <= long_diff <= 0.009 and -0.009 <= lat_diff <= 0.009:
            flag = True
        else:
            flag = False

        # print("Comparing data from Dataset 2 ...")
        if -0.009 <= long_diff4 <= 0.009 and -0.009 <= lat_diff4 <= 0.009:
            flag4 = True
        else:
            flag4 = False

        # print("Comparing data from Dataset 3 ...")
        if -0.009 <= long_diff5 <= 0.009 and -0.009 <= lat_diff5 <= 0.009:
           flag5 = True
        else:
           flag5 = False



if flag == True:
      print("Coordinates Lie in Range for Dataset 1")
else:
        print("Heads up... Difference detected for Dataset 1")

if flag4 == True:
      print("Coordinates Lie in Range for Dataset 2")
else:
        print("Heads up... Difference detected for Dataset 2")

if flag5 == True:
      print("Coordinates Lie in Range for Dataset 3")
else:
        print("Heads up... Difference detected for Dataset 3")
```

Figure 5.10 Python Code for Comparison over Encrypted Data

The display of results on the user's screen provides real-time feedback regarding the comparison between the user-specified batch data and the reference datasets as shown in Figure 5.11. This enables vessel staff or users to quickly assess the conformity of the vessel's route and take appropriate measures if any deviations are detected.



```
E:\Work\python\Latest Implementation pyope\new>python comparison.py
Database Connected!
Enter the port Locode from where you want to start your journey: pkkhi
Start Port is: pkkhi
Enter the port Locode where you want to end your journey: mypkg
End Port is: mypkg
Enter batch of data you want to do comparison for: 25
It took 3.52 seconds for Encryption of 25 batch of data
Coordinates Lie in Range for Dataset 1
Heads up... Difference detected for Dataset 2
Coordinates Lie in Range for Dataset 3
It took 4.59 seconds for Comparison of 25 batch of data
```

Figure 5.11 PYOPE Implementation UI

5.3 Results and Analysis

Based on the tests conducted, a comprehensive analysis of the collected data has yielded the following results:

5.3.1 Input Output Analysis

The Order Preserving Encryption (OPE) algorithm ensures that the numerical ordering of the plaintext values is preserved during the encryption process. This means that the corresponding ciphertext values will maintain the same ordering as the original plaintext values. For example, the plaintext value of 10 is encrypted using the OPE library with randomly generated key results in a ciphertext value of 909089. The increase in the size of the ciphertext from 4 bits to 20 bits indicates that the encryption process introduces additional information to effectively preserve the ordering of the data. This expansion in size is necessary to ensure that the encrypted values maintain their proper sequence when decrypted. Table 5.1 indicates the same.

Sr. No.	**Input(PT)**	**No. of Input Bits**	**Output(CT)**	**No. of Output Bits**	**Encryption Time(s)**	**Decryption Time(s)**
1	10	4	909089	20	0.097	0.003
2	100	7	7009492	23	0.097	0.004
3	1000	10	69089039	27	0.098	0.005
4	10000	14	648128292	30	0.097	0.005
5	99999	17	2744338500336	37	0.113	0.019
6	9999999	24	343484236220	40	0.112	0.018
7	11393579101562	44	193531668104276	48	0.104	0.017
8	24825510978699	45	421228500592632	49	0.122	0.034
9	66980350494385	46	1136346201403111	51	0.113	0.041
10	1000000000000000	50	16998926889729138	54	0.123	0.019

Table 5. 1 PT CT Comparison in terms of PYOPE

The encryption time of 0.097 seconds and decryption time of 0.003 seconds provide insights into the efficiency of the encryption and decryption operations using the OPE library. These timings indicate the amount of time required to encrypt and decrypt the data, respectively. The relatively fast encryption and decryption times suggest that the OPE algorithm implemented in the library is efficient in terms of processing speed.

5.3.2 Encryption Analysis

We conducted a series of tests using the designed code, keeping the encryption key fixed, and obtained the following results.

The process of adding 500 meters to five latitude-longitude pairs, performing floating point adjustment, and applying PYOPE encryption takes a total time of 0.71 seconds. Performing same steps for fifteen latitude-longitude pairs takes a total time of approximately 2.73 seconds. This duration includes the necessary computations and operations involved in each step of the process. For forty pairs, the processing time increases to around 12.81 seconds. With forty-five pairs, it further increases to approximately 16.07 seconds. Finally, when handling fifty pairs, the processing time reaches around 19.49 seconds. It is to be noted that these times are estimates and can vary based on various factors such as hardware capabilities, system load, and implementation specifics. The actual execution time might slightly differ in different environments or with different datasets.

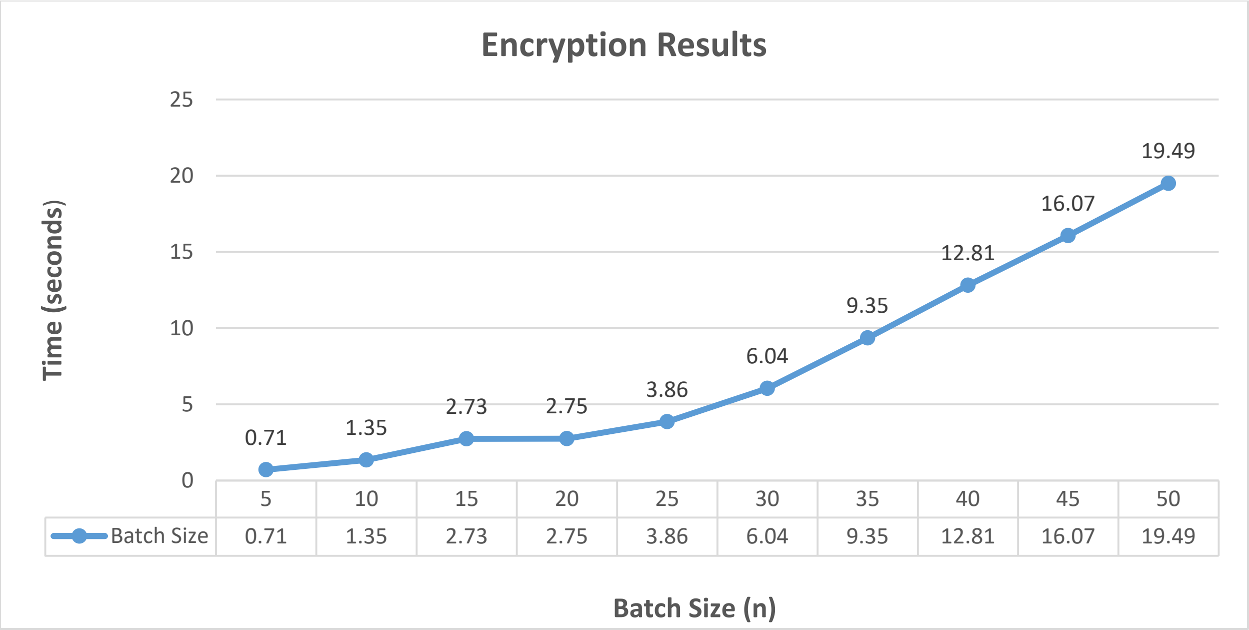


Figure 5.12 Graph showing Encryption Results and Trend

Figure 5.12 illustrates encryption results and trend where n represents number of latitude, longitude pairs.

Based on the provided pairs and their corresponding processing times, we can observe the Time Increase with Pair Count Trend. With an increase in the quantity of latitude-longitude pairs, there is a corresponding rise in the processing time. This is expected since each additional pair requires additional calculations and encryption operations. The relationship between pair count and processing time appears to be roughly linear, with a gradual increase in time as more pairs are processed.

5.3.3 Decryption Results

Based on a series of decryption tests, we have obtained the following findings:

1. Increasing Batch Size: As the batch size increases, there is a gradual increase in the time taken for the decryption process. This suggests that the decryption time is directly proportional to the batch size. The larger the batch size, the more time-consuming the decryption becomes.

2. Linear Relationship: The relationship between the batch size and decryption time appears to be approximately linear. As the batch size doubles, the decryption time roughly doubles as well. This implies a consistent decryption speed across different batch sizes.
3. Efficient Decryption: The decryption process demonstrates efficient performance overall, with relatively low decryption times even for larger batch sizes. The process remains reasonably fast and suitable for practical applications, as the decryption time for all batch sizes remains within a few seconds.
4. Scalability: The decryption process exhibits good scalability, as the time increase with larger batch sizes is relatively proportional. This suggests that the decryption algorithm can handle larger data sets without a significant degradation in performance.

Based on these findings, it can be concluded that the decryption process is efficient and capable of handling various batch sizes effectively. Figure 5.13 demonstrates the trend of decryption results by presenting a graph that plots the batch size (n), n representing the number of latitude and longitude pairs, against time in seconds.

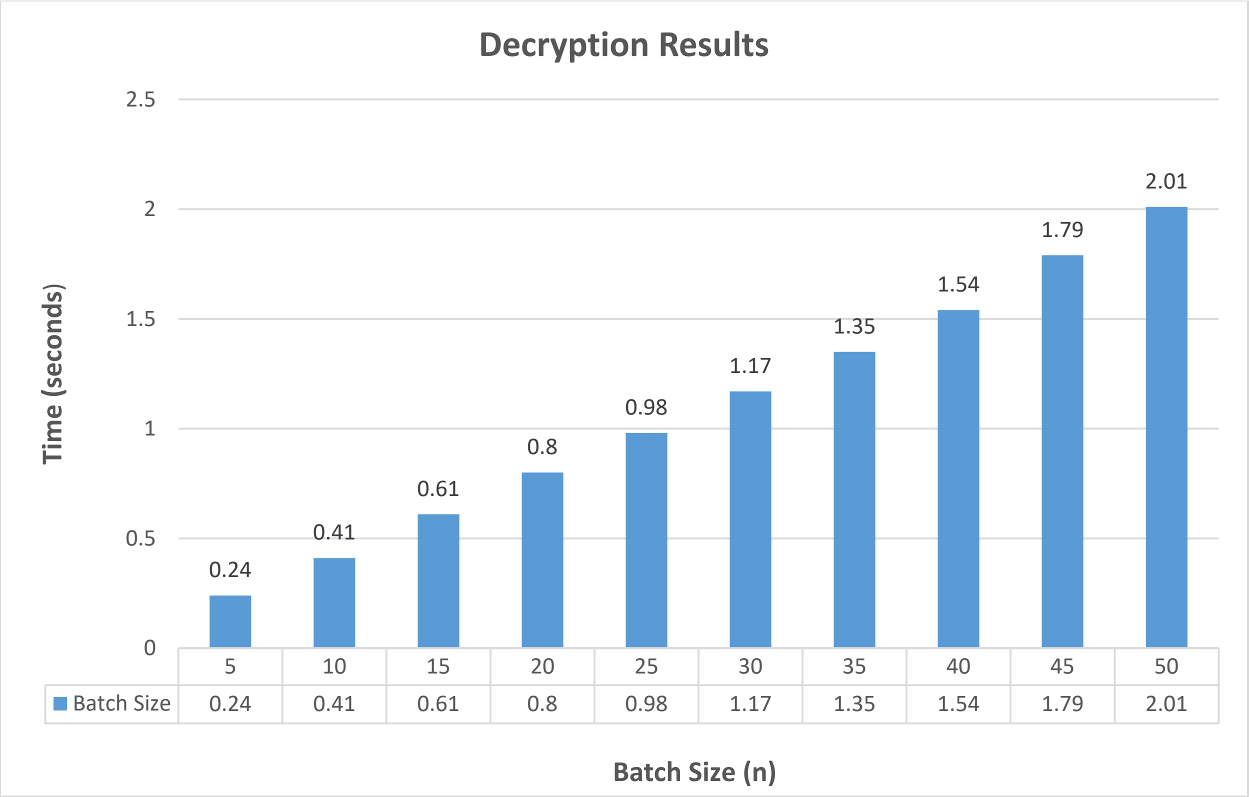


Figure 5.13 Graph showing Decryption Results and Trend

5.3.4 Comparison Results

To evaluate the comparison section of the code, we conducted tests on data sets of varying batch sizes (represented by 'n'). The results of these tests are depicted in Figure 5.14.

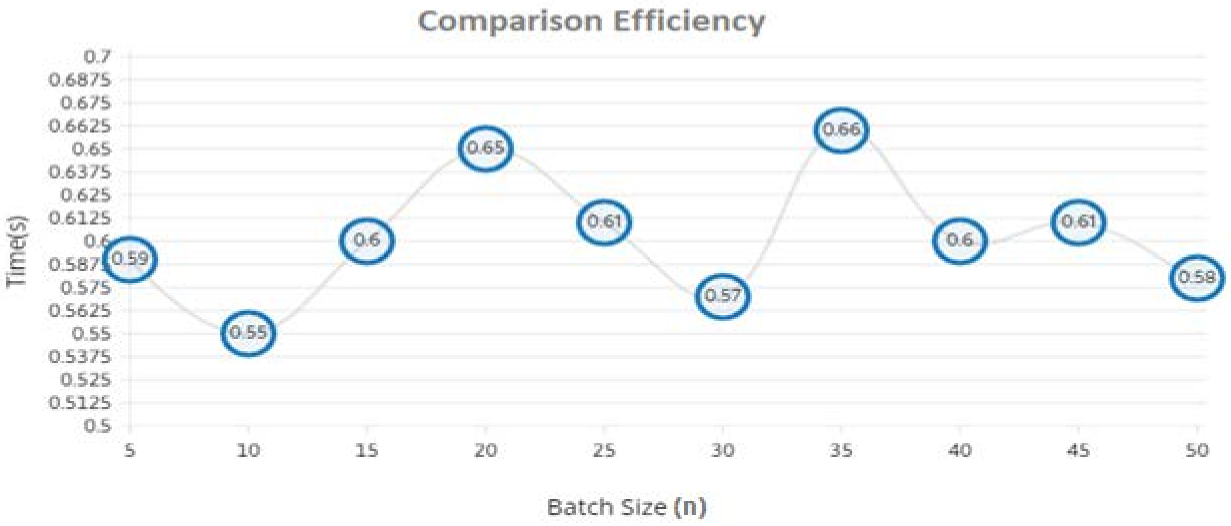


Figure 5.14 Comparison Efficiency Graph

Based on the results obtained from the comparison phase of the code, we can observe the following trends:

1. Time Stability: The time taken for the comparison process shows relatively stable performance across different batch sizes. There is no significant increase or decrease in the time as the batch size increases.
2. Consistency: The time measurements for most batch sizes range between 0.55 to 0.66 seconds. This indicates that the code consistently performs the comparison task within a relatively narrow time range.
3. Efficiency: Overall, the comparison process demonstrates efficient performance as the batch size increases. The time taken for the comparison remains relatively low, suggesting that the code can handle larger batch sizes without significant performance degradation.
4. Scalability: The code exhibits good scalability as the batch size increases. There is no significant increase in the time required for comparisons, indicating that the code can handle larger datasets efficiently.
5. Optimization Potential: The consistency in time measurements suggests that the code may have reached a point of optimization, where further improvements may not result in significant time savings. However, if faster processing is desired, exploring potential optimization techniques could be considered.

Overall, the analysis indicates that the code performs well in terms of efficiency and scalability during the comparison phase, providing reliable and consistent results within a reasonable time frame for various batch sizes of data.

5.3.5 Combined Performance Analysis

The performance analysis reveals the duration of various processes, including data retrieval, modification, encryption, and comparison, for different batch sizes (represented by the number of latitude, longitude pairs, denoted as 'n'). Figure. 5.15 visually illustrates the outcomes.

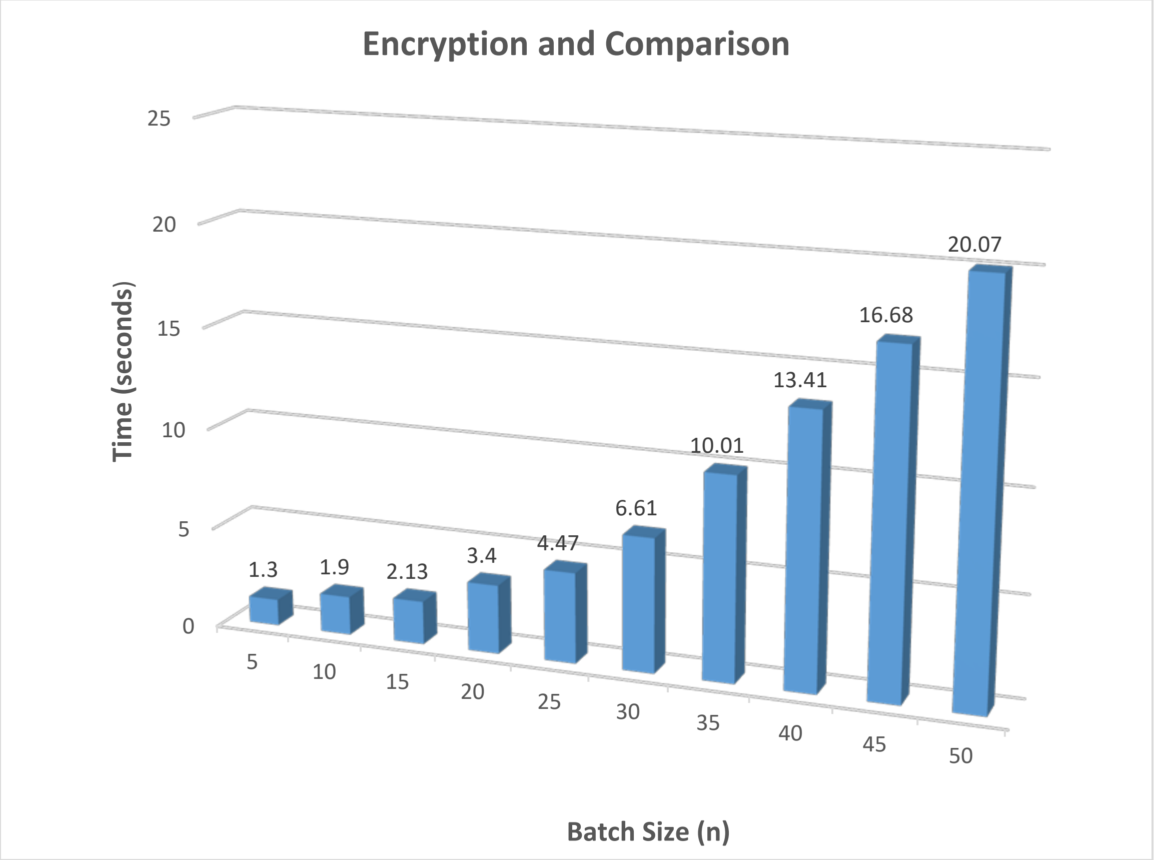


Figure 5.15 Performance Analysis of Code Execution

Analyzing the performance results, we can observe the following trends:

1. Increasing Batch Size: As the batch size increases from 5 to 50, we can see a general trend of the processing time gradually increasing. This suggests that larger batch sizes require more time for data retrieval, modification, encryption, and comparison operations.

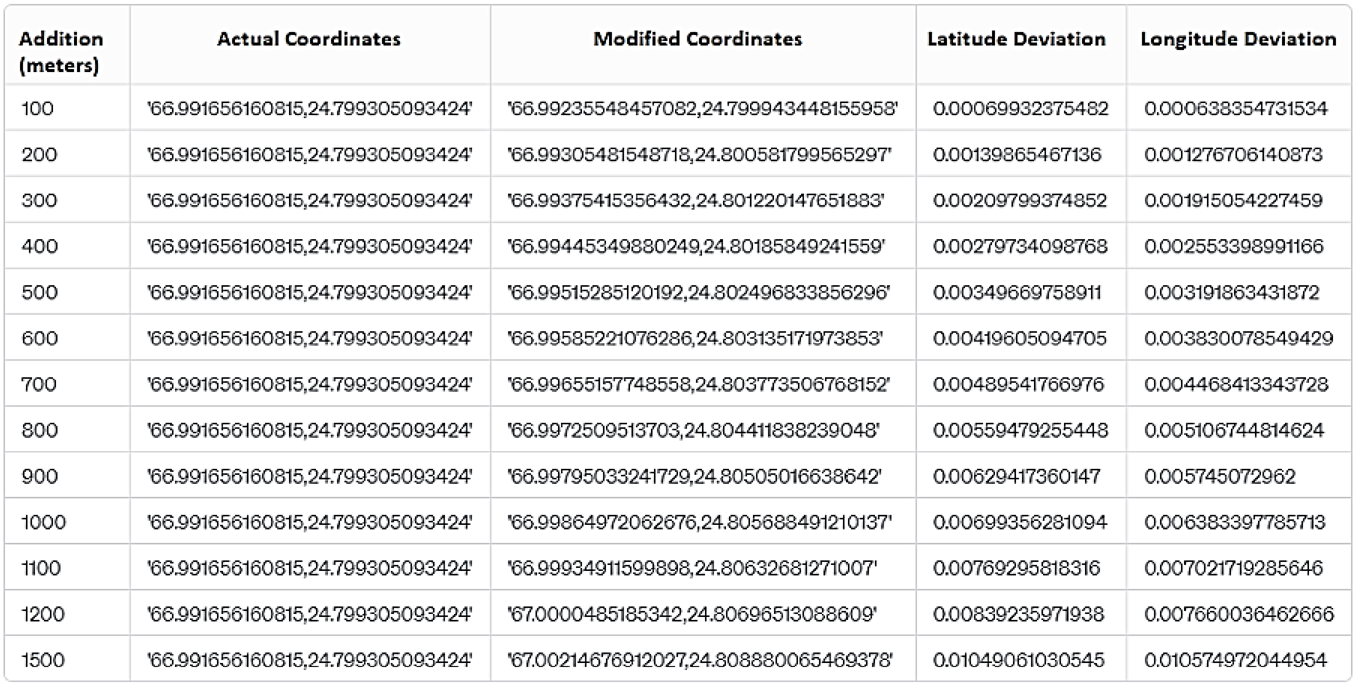
2. Linear Relationship: The relationship between batch size and processing time appears to be roughly linear. As the batch size doubles (e.g., from 5 to 10), the processing time also roughly doubles. This indicates that the code's performance scales proportionally with the batch size.

3. Processing Time Variation: Although there is a general increasing trend, there are slight variations in processing time for different batch sizes. For example, the processing time for a batch size of 15 is slightly higher than that for a batch size of 10. These variations

are influenced by factors such as the complexity of data retrieval, the computational overhead of encryption, or system resource utilization.

5.3.6 Analysis of Coordinate Deviations with 500-Meter Offset

The table analysis supports the justification for using a 500-meter offset in terms of balancing sensitivity to deviations, maintaining proximity to the original data, and ensuring accurate and reliable comparisons. The chosen offset value demonstrates a reasonable compromise between capturing meaningful route variations and avoiding false positives or excessive displacement.



Addition (meters)	Actual Coordinates	Modified Coordinates	Latitude Deviation	Longitude Deviation
100	'66.991656160815,24.799305093424'	'66.99235548457082,24.799943448155958'	0.00069932375482	0.000638354731534
200	'66.991656160815,24.799305093424'	'66.99305481548718,24.800581799565297'	0.00139865467136	0.001276706140873
300	'66.991656160815,24.799305093424'	'66.99375415356432,24.801220147651883'	0.00209799374852	0.001915054227459
400	'66.991656160815,24.799305093424'	'66.99445349880249,24.80185849241559'	0.00279734098768	0.002553398991166
500	'66.991656160815,24.799305093424'	'66.99515285120192,24.802496833856296'	0.00349669758911	0.003191863431872
600	'66.991656160815,24.799305093424'	'66.99585221076286,24.803135171973853'	0.00419605094705	0.003830078549429
700	'66.991656160815,24.799305093424'	'66.99655157748558,24.803773506768152'	0.00489541766976	0.004468413343728
800	'66.991656160815,24.799305093424'	'66.9972509513703,24.804411838239048'	0.00559479255448	0.005106744814624
900	'66.991656160815,24.799305093424'	'66.99795033241729,24.80505016638642'	0.00629417360147	0.005745072962
1000	'66.991656160815,24.799305093424'	'66.99864972062676,24.805688491210137'	0.00699356281094	0.006383397785713
1100	'66.991656160815,24.799305093424'	'66.99934911599898,24.80632681271007'	0.00769295818316	0.007021719285646
1200	'66.991656160815,24.799305093424'	'67.0000485185342,24.80696513088609'	0.00839235971938	0.007660036462666
1500	'66.991656160815,24.799305093424'	'67.00214676912027,24.808880065469378'	0.01049061030545	0.010574972044954

Table 5. 2 Coordinate Deviations with Offset Addition

Analyzing the Table 5.2 in the context of the justification for using a 500-meter offset, we can observe the following:

1. Deviations: The table displays the deviations between the actual coordinates and the modified coordinates for different offset values ranging from 100 to 1500 meters. As the offset increases, the deviations generally become larger, indicating a greater displacement from the original coordinates.

2. Sensitivity to Deviations: The table demonstrates that increasing the offset beyond 500 meters leads to more significant deviations. For example, as the offset increases from 500

to 1500 meters, the deviations become noticeably larger. This aligns with the justification that a higher offset could potentially move the coordinates too far away from their original positions, potentially leading to false positives or inaccuracies in the comparison process.

3. Proximity to Original Data: Despite the increasing deviations, the modified coordinates with a 500-meter offset still maintain a reasonable proximity to the actual coordinates. This reflects the intent to introduce a noticeable change while retaining a recognizable relationship to the original sea route. The deviations within the range of 500 meters generally indicate a reasonable displacement that can effectively capture small variations in the vessel's actual path.

4. Balance of Accuracy: The chosen offset of 500 meters strikes a balance between sensitivity to deviations and accuracy in the comparison process. It provides a level of displacement that is distinguishable from the original data, allowing for effective detection of meaningful deviations from the expected route. At the same time, it avoids excessive displacement that could lead to false positives or inaccurate comparisons.

Throughout the analysis conducted, we explored various aspects of the sea route comparison system. We examined the process of retrieving data from a database, applying modifications such as adding a 500-meter offset, performing encryption using the PYOPE library, and conducting comparisons between datasets.

The performance analysis revealed that the processing time increased as the batch size or the number of latitude-longitude pairs grew larger. The encryption and comparison operations showed consistent time durations across different batch sizes.

Additionally, the analysis of coordinate deviations showcased the impact of adding a 500-meter offset to the latitude-longitude pairs. The modified coordinates exhibited noticeable differences from the original values while still maintaining a reasonable proximity.

Overall, these findings provide insights into the performance and effectiveness of the sea route comparison system, demonstrating its ability to handle data retrieval, modification, encryption, and comparison operations.

Chapter 6

Conclusion and Future Work

In open areas, where ships operate with their substantial size and slow movement (15-25 knots), the absence of interference from other targets makes spoofing a ship relatively simple. In such environments, with limited satellite observability and the absence of reference paths, the slow speed of the target facilitates spoofing. Spoofing can be achieved through hardware injection or by utilizing an escorting operating location or a stand-off approach. However, when it comes to spoofing or jamming the precise location of fishing vessels in prohibited waters, the potential losses can be significant. The spoofer finds it relatively easy to track the ship due to factors such as the ship's larger Radar Cross Section (RCS), the absence of multiple targets in the surveillance area, and the increased variations in pitch and yaw caused by ocean flow[54].

Our research work culminated in the development of a comprehensive framework for detecting GPS spoofing in maritime environments. The proposed framework incorporates a hybrid scheme of encryption techniques and employs an extensive comparison process to enhance the accuracy and reliability of the detection mechanism.

6.1 Research Overview and Conclusion

In summary, our work involved the utilization of searoutes.com API for collecting coordinates of the vessel's intended route. The collected sea route data was encrypted using the

OPE encryption technique and stored in a database. We developed a user-friendly graphical user interface (UI) that allowed users to input the source and destination ports, as well as the batch of data for comparison. Real-time GPS data collection was not feasible, so we gathered three distinct sea route datasets at different time intervals for comparison.

The code retrieved the reference data for the user-specified batch from the datasets and performed a comparison by calculating differences in latitude and longitude values. If the differences fell within a predefined buffer range, indicating adherence to the expected route, a message of compliance was displayed. If the differences exceeded the buffer range, suggesting a significant deviation, a warning message was shown.

The results were displayed in real-time on the user's screen, providing immediate feedback on the conformity of the vessel's route. This allowed for prompt actions to be taken in case of deviations. The analysis of the collected data sets provided valuable insights into the vessel's trajectory, although hardware implementation was beyond the scope of our research.

The diligent research and thorough analysis of the outcomes have produced the following findings:

6.1.1 Route Conformance

The comparison between the user-specified batch data and the reference datasets indicates the level of conformance of the vessel's actual route with the expected route. The results highlight whether the vessel has followed the intended path or deviated from it. This information helps assess the adherence to the planned route and ensures navigational accuracy.

6.1.2 Deviation Detection

By evaluating the differences between latitude and longitude values, the code identifies instances of significant deviations from the expected route. These deviations may occur due to factors such as weather conditions or navigational errors. The results highlight the magnitude and frequency of such deviations, allowing for prompt corrective actions to be taken.

6.1.3 Error Detection

The code also incorporates error handling mechanisms to identify and report any inconsistencies or invalid inputs provided by the user. If incorrect locodes or routes that are not present in the database are entered, appropriate error messages are displayed on the user's screen. This helps ensure the reliability and validity of the comparison process.

6.1.4 Real-time Feedback

The displayed results provide real-time feedback to the user, enabling them to make informed decisions based on the comparison outcomes. The feedback includes clear indications of route compliance, potential deviations, and warnings for necessary actions. This real-time information enhances situational awareness and facilitates timely response in case of route discrepancies.

6.1.5 Data Integrity

The comparison process ensures the integrity of the collected data by verifying its accuracy and legitimacy. By retrieving and comparing data from multiple datasets, the code enhances the reliability of the results and minimizes the impact of potential outliers or anomalies in individual datasets.

6.1.6 Data Confidentiality

By employing encryption at every stage i.e. storage, retrieval, and comparison, the framework ensures the utmost confidentiality of the sea route data. This approach mitigates the risk of unauthorized access to the sensitive information, providing a secure environment for data handling and analysis.

Overall, the research work conducted has laid the foundation for detecting maritime GPS spoofing using OPE and Cloud system and has provided valuable insights into securing sea route data. The framework, with its encryption techniques and comparison process, serves as a powerful tool for ensuring the integrity and reliability of maritime navigation systems.

6.2 Future Work

In light of the results obtained from this research, there are some avenues for future exploration and advancement in the field.

a) Hardware Implementation

Exploring the integration of hardware components and real-time GPS data collection methods can enhance the accuracy and efficiency of the system.

b) Improved Encryption Algorithm

It is to acknowledge that the OPE encryption scheme, despite its ability to maintain data order, necessitates a careful evaluation of its limitations and security implications. The algorithm's resilience against potential vulnerabilities, such as information leakage[55], can be addressed to maximize the benefits derived from the proposed technique

c) Time Optimization

Processing time's analysis for some operations suggests optimization. Methods for enhancing the efficiency of the operations, including refining encryption algorithm, leveraging parallel processing[56], and implementing hardware acceleration can be explored.

d) Integration and Collaboration

Opportunities for integrating the developed code and processes into larger systems or frameworks can be explored. This can involve seeking collaborations with domain experts, researchers, or organizations within the maritime industry. By leveraging their expertise, accessing additional data sources, and contributing to real-world applications, the overall effectiveness and applicability of the proposed framework can be significantly enhanced. Collaboration with relevant stakeholders can also help in validating the system's performance, addressing potential limitations, and ensuring its seamless integration into existing maritime systems.

BIBLIOGRAPHY

[1] Chen, C., Shiotani, S., & Sasa, K. (2013). Numerical ship navigation based on weather and ocean simulation. Ocean Engineering, 69, 44-53.

[2] Nilsson, R., Gärling, T., & Lützhöft, M. (2009). An experimental simulation study of advanced decision support system for ship navigation. Transportation research part F: traffic psychology and behaviour, 12(3), 188-197.

[3] "Navigation", education.nationalgeographic.org https://education.nationalgeographic.org/resource/navigation/ (accessed June 09, 2023)

[4] Hegarty, C. J. (2017). The global positioning system (GPS). Springer Handbook of Global Navigation Satellite Systems, 197-218.

[5] Hofmann-Wellenhof, B., Lichtenegger, H., & Collins, J. (2012). Global positioning system: theory and practice. Springer Science & Business Media.

[6] Bajaj, R., Ranaweera, S. L., & Agrawal, D. P. (2002). GPS: location-tracking technology. Computer, 35(4), 92-94.

[7] Dahmann, J. S., Fujimoto, R. M., & Weatherly, R. M. (1997, December). The department of defense high level architecture. In Proceedings of the 29th conference on Winter simulation (pp. 142-149).

[8] Kent, S. (1991). US department of defense security options for the internet protocol (No. rfc1108).

[9] Andrew Young, Christina Rogawski and Stefaan Verhulst, “United States Opening GPS for Civilian Use”, oreilly.com https://www.oreilly.com/library/view/the-global-impact/9781492042785/ch15.html (accessed June 09, 2023)

[10] “How Does GPS Work?”, nasa.gov https://spaceplace.nasa.gov/gps/en/ (accessed June 11, 2023)

[11] Khan, S. Z., Mohsin, M., & Iqbal, W. (2021). On GPS spoofing of aerial platforms: a review of threats, challenges, methodologies, and future research directions. PeerJ Computer Science, 7, e507.

[12] Peng Jiang, Hongyi Wu, Chunsheng Xin, DeepPOSE: Detecting GPS spoofing attack via deep recurrent neural network, Digital Communications and Networks, Volume 8, Issue 5, 2022, Pages 791-803, ISSN 2352-8648

[13] Michael Jones, “Spoofing in the Black Sea: What really happened?”, gpsworld.com https://www.gpsworld.com/spoofing-in-the-black-sea-what-really-happened/ (accessed June 11, 2023)

[14] “Mass global positioning system spoofing at ports in PRC”, insurancemarinenews.com https://insurancemarinenews.com/insurance-marine-news/mass-global-positioning-system-spoofing-at-ports-in-prc/ (accessed June 11, 2023)

[15] Juliana De Groot “What Is Data Encryption? (Definition, Best Practices & More)”, digitalguardian.com https://www.digitalguardian.com/blog/what-data-encryption (accessed June 11, 2023)

[16] Xiao, L., & Yen, I. L. (2012). A note for the ideal order-preserving encryption object and generalized order-preserving encryption. Cryptology ePrint Archive.

[17] “Guide to Maritime Security”, mitags.org https://www.mitags.org/security-guide/ (accessed June 02, 2023)

[18] Liwång, H., Sörenson, K. & Österman, C. Ship security challenges in high-risk areas: manageable or insurmountable?. WMU J Marit Affairs 14, 201–217 (2015).

[19] Alcaide, Juan & Garcia-Llave, Ruth. (2020). Critical infrastructures cybersecurity and the maritime sector. Transportation Research Procedia. 45. 547-554. 10.1016/j.trpro.2020.03.058.

[20] Gu, Y., Goez, J. C., Guajardo, M., & Wallace, S. W. (2021). Autonomous vessels: state of the art and potential opportunities in logistics. International Transactions in Operational Research, 28(4), 1706-1739.

[21] Ben Farah, M. A., Ukwandu, E., Hindy, H., Brosset, D., Bures, M., Andonovic, I., & Bellekens, X. (2022). Cyber security in the maritime industry: A systematic survey of recent advances and future trends. Information, 13(1), 22.

[22] Menhat, M., Zaideen, I. M. M., Yusuf, Y., Salleh, N. H. M., Zamri, M. A., & Jeevan, J. (2021). The impact of Covid-19 pandemic: A review on maritime sectors in Malaysia. Ocean & coastal management, 209, 105638.

[23] Larsen, M. H., & Lund, M. S. (2021). A maritime perspective on cyber risk perception: A systematic literature review. IEEE Access, 9, 144895-144905.

[24] Kavallieratos, G., Katsikas, S., & Gkioulos, V. (2019). Cyber-attacks against the autonomous ship. In Computer Security: ESORICS 2018 International Workshops, CyberICPS 2018 and SECPRE 2018, Barcelona, Spain, September 6–7, 2018, Revised Selected Papers 2 (pp. 20-36). Springer International Publishing.

[25] Svilicic, B., Brčić, D., Žuškin, S., & Kalebić, D. (2019). Raising awareness on cyber security of ECDIS. TransNav: International Journal on Marine Navigation and Safety of Sea Transportation, 13(1), 231-236.

[26] Svilicic, B., Kamahara, J., Celic, J., & Bolmsten, J. (2019). Assessing ship cyber risks: A framework and case study of ECDIS security. WMU Journal of Maritime Affairs, 18, 509-520.

[27] Kavallieratos, G., Diamantopoulou, V., & Katsikas, S. K. (2020). Shipping 4.0: Security requirements for the cyber-enabled ship. IEEE Transactions on Industrial Informatics, 16(10), 6617-6625.

[28] Heffner, C. “Exploiting Surveillance Cameras Like a Hollywood Hacker”, privacy-pc.com https://privacy-pc.com/articles/exploiting-network-surveillance-cameras-like-a-hollywood-hacker.html (accessed on 5 June 2023).

[29] Bugeja, J., Jönsson, D., & Jacobsson, A. (2018, March). An investigation of vulnerabilities in smart connected cameras. In 2018 IEEE international conference on pervasive computing and communications workshops (PerCom workshops) (pp. 537-542). IEEE.

[30] Precise Positioning in Urban Canyons: Applied to the Localisation of Buried Assets - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Overview-of-the-GPS-segments-USArmy-1996_fig3_325626210 [accessed 15 May, 2023]

[31] Altimetry with GNSS Bistatic Radar - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/GPS-signal-generation_fig1_281652888 [accessed 15 May, 2023]

[32] Cloud and MEC security - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/A-view-of-the-cloud-architecture_fig1_322467223 [accessed 15 May, 2023]

[33] Boldyreva, Alexandra & Chenette, Nathan & Lee, Younho & O'Neill, Adam. (2009). Order-Preserving Symmetric Encryption. Advances in Cryptology-EUROCRYPT 2009. 5479. 224-241. 10.1007/978-3-642-01001-9_13.

[34] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Order-preserving encryption for numeric data. In SIGMOD ’04, pp. 563–574. ACM, 2004.

[35] Spravil, J.; Hemminghaus, C.; von Rechenberg, M.; Padilla, E.; Bauer, J. Detecting Maritime GPS Spoofing Attacks Based on NMEA Sentence Integrity Monitoring. J. Mar. Sci. Eng. 2023, 11, 928. https://doi.org/10.3390/jmse11050928

[36] Singh S, Singh J, Singh S, Goyal SB, Raboaca MS, Verma C, Suciu G. Detection and Mitigation of GNSS Spoofing Attacks in Maritime Environments Using a Genetic Algorithm. Mathematics. 2022; 10(21):4097. https://doi.org/10.3390/math10214097

[37] Abreu, F.H.O.; Soares, A.; Paulovich, F.V.; Matwin, S. A Trajectory Scoring Tool for Local Anomaly Detection in Maritime Traffic Using Visual Analytics. ISPRS Int. J. Geo-Inf. 2021, 10, 412. https://doi.org/10.3390/ijgi10060412

[38] Androjna, Andrej & Perkovic, Marko. (2021). Impact of Spoofing of Navigation Systems on Maritime Situational Awareness. Transactions on Maritime Science. 10. 10.7225/toms.v10.n02.w08.

[39] Bhatti, Jahshan & Humphreys, Todd. (2017). Hostile Control of Ships via False GPS Signals: Demonstration and Detection. Navigation. 64. 51-66. 10.1002/navi.183.

[40] Ben Farah, M.A.; Ukwandu, E.; Hindy, H.; Brosset, D.; Bures, M.; Andonovic, I.; Bellekens, X. Cyber Security in the Maritime Industry: A Systematic Survey of Recent Advances and Future Trends. Information 2022, 13, 22. https://doi.org/10.3390/info13010022

[41] Dana, Peter H.. "Global Positioning System (GPS) Time Dissemination for Real-Time Applications." Real-Time Systems 12 (1997): 9-40.

[42] Wullems, Christian & Pozzobon, Oscar & Kubik, Kurt. (2005). Signal Authentication and Integrity Schemes for Next Generation Global Navigation Satellite Systems.

[43] Schmidt, Desmond & Radke, Kenneth & Camtepe, Seyit & Foo, Ernest & Ren, Michał. (2016). A Survey and Analysis of the GNSS Spoofing Threat and Countermeasures. ACM Computing Surveys. 48. 1-31. 10.1145/2897166.

[44] Kuhn, Markus. (2004). An Asymmetric Security Mechanism for Navigation Signals. 3200. 10.1007/978-3-540-30114-1_17.

[45] Montgomery, Paul Y., Todd E. Humphreys, and Brent M. Ledvina. "Receiver-autonomous spoofing detection: Experimental results of a multi-antenna receiver defense against a portable civil GPS spoofer." Proceedings of the 2009 International Technical Meeting of The Institute of Navigation. 2009.

[46] Tippenhauer, Nils Ole & Pöpper, Christina & Rasmussen, Kasper & Capkun, Srdjan. (2011). On the requirements for successful GPS spoofing attacks. 75-86. 10.1145/2046707.2046719.

[47] Wang, Qian, et al. "Edge computing based GPS spoofing detection methods." 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP). IEEE, 2018.

[48] Lakebed Characterization Using Side-Scan Data for Investigating the Latest Lake Superior Coastal Environment Conditions - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/RTK-GPS-Method-for-Maritime-Projects-US-Geological-Survey-Department-of-the_fig15_322739954 [accessed 15 May, 2023]

[49] "Mass Global Positioning System (GPS) Spoofing At Ports In The People's Republic Of China", hellenicshippingnews.com. https://www.hellenicshippingnews.com/mass-

global-positioning-system-gps-spoofing-at-ports-in-the-peoples-republic-of-china/ (accessed May. 15, 2023).

[50] Chris Lo, “GPS spoofing: what’s the risk for ship navigation?”, ship-technology.com https://www.ship-technology.com/features/ship-navigation-risks/ (accessed May. 16, 2023).

[51] “Searoutes - Making supply chains greener.”, searoutes.com https://searoutes.com/ (accessed May. 17, 2023).

[52] Anton Ovchinnikov , “Implementation of symmetric order-preserving encryption scheme”, pypi.org https://pypi.org/project/pyope/(accessed May. 18, 2023)

[53] “2022-005-Various-GPS Interference & AIS Spoofing”, maritime.dot.gov https://maritime.dot.gov/msci/2022-005-various-gps-interference-ais-spoofing/(accessed May. 20, 2023)

[54] P. Bethi, S. Pathipati and A. P, "Stealthy GPS Spoofing: Spoofer Systems, Spoofing Techniques and Strategies," 2020 IEEE 17th India Council International Conference (INDICON), New Delhi, India, 2020, pp. 1-7, doi: 10.1109/INDICON49873.2020.9342317.

[55] Chenette, N., Lewi, K., Weis, S. A., & Wu, D. J. (2016). Practical order-revealing encryption with limited leakage. In Fast Software Encryption: 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers 23 (pp. 474-493). Springer Berlin Heidelberg.

[56] Margaret Rouse, “Parallel Processing”, techopedia.com https://www.techopedia.com/definition/4598/parallel-processing (accessed June. 11, 2023)