

**DESIGN AND EVALUATION OF ATTRIBUTE-BASED ENCRYPTION
FOR DATA SECURITY**



Author

Hamza Safdar

0000329369

Supervisor

Dr. Bilal Muhmmad Khan

THESIS

Submitted to:

Department of Cybersecurity

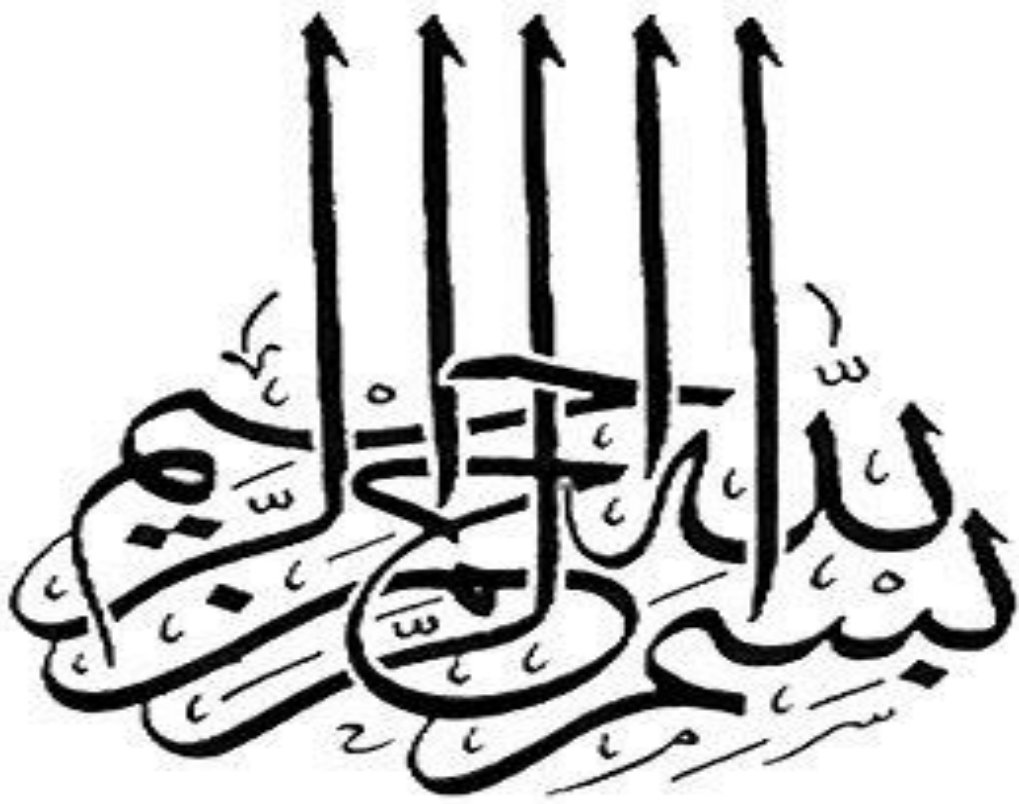
Pakistan Navy Engineering College, Karachi

National University of Sciences and Technology, Islamabad, Pakistan

In partial fulfillment of requirements for the award of the degree of

MASTER OF SCIENCE IN CYBER SECURITY

March 2024




National University of Sciences and Technology


MASTER'S THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by: (Student Name & Regn No.) Hamza Safdar (00000329396) Titled: DESIGN AND EVALUATION OF ATTRIBUTE BASED ENCRYPTION FOR DATA SECURITY be accepted in partial fulfillment of the requirements for the award of Master's degree.

EXAMINATION COMMITTEE MEMBERS

1. Name: Dr. Lubna Moin Signature: 

2. Name: Dr. Rashida A. Memon Signature: 

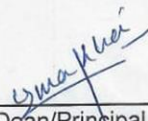
Supervisor's name: Dr. Bilal Muhammad Khan Signature: 
Date: 28-03-2024

28-03-2024
Date:


ALIYA ALI
Lt Cdr Pakistan Navy
Head of Department

COUNTERSIGNED


Date: 28-03-2024



Dean/Principal


UZMA KHALID
Cdr Pakistan Navy
HOD Computer Science
PNS Jauhar

THESIS ACCEPTANCE CERTIFICATE

Certified that the final copy of the MS thesis written by HAMZA SAFDAR Regn No. 00000329396 of NUST- PNEC (College) has been vetted by the undersigned, found complete in all respects as per NUST Status/Regulations, 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 of the scholar have been incorporated in the said thesis.

Signature:  _____
Name of Supervisor: DR. BILAL MUHAMMAD KHAN
Dr. Bilal Muhammad Khan
Director R&D
Dated: 28-03-2021 NUST-PNEC

Signature: HoD  _____
Dated: 28-03-2021 dr Pakistan Navy
HOD CySD

Signature: (Dean/Principal):  _____
Dated: 28-03-2021 Uzma Khalid
dr Pakistan Navy
HOD Computer Science
PNS Jauhar

Approval

It is certified that the contents and form of the thesis entitled "Design and Evaluation of Attribute-Based Encryption for Data Security" submitted by "Hamza Safdar" have been found satisfactory for the requirement of the degree.

Name of Supervisor Dr. Bilal Muhammad Khan

Signature:  _____

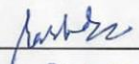
Date: 28-03-2024

Committee Member: Dr. Lubna Moin

Signature:  _____

Date: 28-03-2024

Committee Member: Dr. Rashida A. Memon


Signature:  _____

Date: 28-03-2024

CERTIFICATE FOR PLAGIARISM

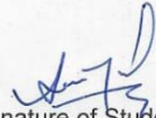
1. It is certified that PhD / M.Phil / **MS** Thesis Titled "**DESIGN AND EVALUATION OF ATTRIBUTE-BASED ENCRYPTION FOR DATA SECURITY**" by **HAMZA SAFDAR(2020-NUST-MS Cyber Security (CyS Fall 20)** has been examined by us. We undertake the follows:

- a. Thesis has significant new work / knowledge as compared to already published or is under consideration to be published elsewhere. No sentence, equation, diagram, table, paragraph or section has been copied verbatim from previous work unless it is placed under quotation marks and duly referenced.
- b. The work presented is original and own work of the author (i.e. there is no plagiarism). No ideas, processes, results or words of others have been presented as Author own work.
- c. There is no fabrication of data or results which have been compiled / analyzed.
- d. There is no falsification by manipulating research materials, equipment, or processes, or changing or omitting data or results such that the research is not accurately represented in the research record.
- e. The thesis has been checked using TURNITIN (copy of originality report attached) and found within limits as per HEC Plagiarism Policy and instructions issued from time to time.


Name & Signature of Supervisor
DR. BILAL MUHAMMAD
Tenured Associate Professor
Director R&D
NUST-PNEC

CERTIFICATE OF ORIGINALITY

I Certify that this research work titled "Design and Evaluation of Attribute-Based Encryption for Data Security" is my own work. The work has not been presented elsewhere for assessment. The material that has been used from the sources has been properly acknowledged/referred.



Signature of Student

Hamza Safdar

2020-NUST-MS-CYS-0000329396

COPYRIGHT STATEMENT

- Copyright in the text of this thesis rests with the student author. Copies (by any process) either in full or of extracts, may be made only in accordance with the instructions given by the author and lodged in the Library of NUST Pakistan Navy Engineering College (PNEC). Details may be obtained by the Librarian. This page must form part of any such copies made. Further copies (by any process) may not be made without the permission (in writing) of the author.
- The ownership of any intellectual property rights that may be described in this thesis is vested in NUST Pakistan Navy Engineering College, subject to any prior agreement to the contrary, and may not be made available for use by third parties without the written permission of the PNEC, which will prescribe the terms and conditions of any such agreement.
- Further information on the conditions under which disclosures and exploitation may take place is available from the Library of NUST Pakistan Navy Engineering College, Karachi.

ACKNOWLEDGMENTS

All praise and thanks to the Almighty ALLAH (SWT) whom we seek help and guidance for sustenance. Firstly, I would like to express my sincere gratitude to my advisor **Dr. Bilal Muhammad Khan** for his patience, motivation, wise reproach, and continuous support of my Master's study and related research. His timely guidance helped me in all the research and writing this thesis. He helps me with his insight and knowledge by enlightening me at the first glance of my research, understanding the subject, and facilitating the fabrication procedure immensely. I could not have imagined having a better advisor and mentor for my Master's study than him hence, I am highly grateful to him for his kind support in all phases of research. Besides my advisor, I would like to thank and appreciate my thesis GEC members: **Dr. Rashida Ali Memon**, and **Dr. Lubna Moin** for their insightful comments and encouragement, but also for the hard questions that encouraged me to widen my research from various perspectives. I would also like to thank the Ex-Head of the Department (PGS-CYSD) **Lt Cdr. Qurratulain Salman** and the present Head of the Department **Lt Cdr. Aliya Ali** for guiding and helping me without their precious support it would not be possible to conduct this research. Last but not least, I would like to thank my family for supporting me financially, spiritually, and in every possible way throughout writing this thesis and my life in general.

ABSTRACT

In an era characterized by the rapid evolution of technology, the paramount importance of safeguarding data security, privacy, and integrity has reached unprecedented levels. While contemporary cryptographic techniques and algorithms have proven to be formidable defenses, the incessantly changing threat landscape necessitates ongoing advancements to effectively combat emerging, often unpredictable threats. Traditional cryptographic methodologies, renowned for their resilience against brute force attacks, find themselves vulnerable to more intricate intrusion techniques employed by malicious actors. These assailants, rather than directly targeting cryptographic elements, often exploit weaknesses inherent to operating systems, harness insider access privileges, manipulate human psychology, or employ social engineering tactics to gain illicit access to network and computer systems.

In response to these ever-evolving challenges, this thesis introduces a novel approach a geo-enhanced encryption technique. This pioneering methodology enriches the well-established Advanced Encryption Standard (AES) algorithm by incorporating user-specific attributes, including geographical location, timestamp data, and other relevant contexts. Furthermore, this approach integrates the use of Pseudo Random-Number Generator (PRNG) and introduces the concept of Toleration Distance (TD), which collectively contributes to an additional layer of security that transcends the confines of conventional cryptographic practices.

The primary objective of this research is to significantly fortify the security, privacy, and integrity of data within various systems, including laptops and mobile devices. By harnessing geo-enhanced encryption, this innovative approach aims to address the limitations of existing cryptographic techniques and effectively counteract the evolving tactics employed by cyber adversaries. Through the incorporation of user-centric attributes and the judicious application of PSEUDO RANDOM NUMBER GENERATOR (PRNG) and TD, this solution promises to augment the overall security posture of systems, allowing for secure access while preserving the confidentiality and integrity of sensitive data. As the digital threat landscape continues to evolve, this thesis serves as a pivotal step toward securing the future of data in an increasingly interconnected and vulnerable world.

Table of Contents

COPYRIGHT STATEMENT	vi
ACKNOWLEDGMENTS	vii
ABSTRACT	viii
LIST OF ACRONYMS	xv
CHAPTER 1: INTRODUCTION	1
1.1 Abstract.....	1
1.2 Research Objectives.....	4
1.3 Thesis Approach	4
1.4 Scope of Research.....	6
1.4.1 Algorithmic Development	6
1.4.2 Experimental Assessment	6
1.4.3 Real-World Applications	7
1.4.4 Interdisciplinary Contributions.....	7
1.4.5 Broader Implications.....	7
1.5 Thesis Challenge	7
1.6 Thesis Organization	9
CHAPTER 2: LITERATURE REVIEW	10
2.1 The Role of Location-Based Encryption.....	10
2.2 Geo-Encryption Algorithm.....	10

2.3	The Foundations of Geo-Encryption	12
2.3.1	Geographic Coordinates: The Cornerstone	12
2.3.2	Time-Based Data: A Dynamic Layer of Security	12
2.3.3	Adaptability in the Face of Modern Threats.....	13
2.3.4	The Broader Implications	13
2.4	Location Dependent Encryption Algorithm (LDEA)	13
2.5	Dynamic Toleration Distance (DTD)	14
2.5.1	Transforming Mobile Node Security.....	15
2.5.2	Dynamic Security Through Mobile Node Movement.....	15
2.5.3	Estimating the Next Position.....	16
2.5.4	Integration of Movement Type	16
2.5.5	Resilience Against Attacks	16
2.5.6	Practical Implications	17
2.6	Use Cases.....	17
2.6.1	Location-based Messaging:	17
2.6.2	Location-based Gaming:	20
2.6.3	Location-Based Advertisement Services:	22
CHAPTER 3: LOCATION-BASED ENCRYPTION		24
3.1.	Overview and Model Portrayal	24
3.1.1.	Overview	24

3.1.2.	Protocol Description	28
3.1.3.	Geographic information utilization during the key derivation process.....	31
3.2.	Detailed Examination of Sender and Receiver Perspectives.....	35
3.3.	Security Requirements	37
3.3.1	Computational Expensiveness.....	37
3.3.2	Nonce Information	39
3.3.3	"Utilizing End-to-End Encryption (E2EE) for Data Security"	40
3.4.	Thread Model	41
3.4.1	Adversaries	41
3.4.2	Key Derivation Function (KDF) Input Space	41
3.4.3	Confidentiality of Communication Channel.....	44
3.4.4	False positive and false negative values	44
3.4.5	Location Verification Claims	47
3.5.	Security Analysis.....	48
3.5.1	Location Authentication	48
3.5.2	Location Disclosure	51
3.5.3	Location Verification	52
CHAPTER 4: PREVENTING GPS/GEO-LOCATION SPOOFING IN ANDROID APPLICATIONS		54
4.1	ANDROID METHODS	55

4.2	Unsupervised Machine Learning	58
4.3	ML Algorithm to Detect Location Spoofing in Android Applications	60
4.3.1	Proposed Solution	60
4.3.2	Implementation and Results	63
4.4	Machine learning models for data analysis	65
4.4.1	Data Preprocessing	65
4.4.1.1	Preprocessing	65
4.4.1.2	Splitting of the data set in Training and Validation sets	66
4.4.1.3	Taking care of Missing values	66
4.4.1.4	Taking Care of Categorical Features	67
4.4.1.5	Normalization of Data Sets	67
4.4.2	Comparison of Different AI Models	67
4.4.2.1	KNN Classifier	67
4.4.2.2	Naive Bayes – Classifier	68
4.4.2.3	Comparison of AI Models.....	68
CHAPTER 5: CONCLUSION AND FUTURE WORK.....		69
5.1	Conclusion.....	69
5.2	Future Recommendations	71
References.....		74

LIST OF FIGURES

FIGURE 1: LOCATION-BASED MESSAGING.....	19
FIGURE 2: LOCATION-BASED GAMING.....	21
FIGURE 3: LOCATION-BASED ADVERTISEMENT.....	23
FIGURE 4: BLOCK DIAGRAM OF THE PROPOSED MODEL.....	27
FIGURE 5: UTILIZATION OF GEO LOCATION IN KDF.....	33
FIGURE 6: OVERVIEW OF THE PROPOSED MODEL.....	36
FIGURE 7: POSSIBLE LOCATION POINTS.....	42
FIGURE 8: LOCATION POINTS ACCURACY.....	46
FIGURE 9: OVERVIEW OF K-MEANS ALGORITHM.....	62

List of Tables

TABLE 1: LIST OF ACRONYMS	XV
TABLE 2: SUMMARY OF THE ACCURACY OF K-MEANS.....	65
TABLE 3: COMPARISON OF AI MODELS.....	68

LIST OF ACRONYMS

Abbreviation	Meaning
AES	Ultra-wideband
LBS:	Location-Based Systems
GPS:	Global Positioning System
CDMA:	Code Division Multiple Access
SA:	Selective Availability
AS:	Anti-spoofing
PNT:	Position, Navigation, and Timing
IMUs:	Inertial Measurement Units
E2EE:	End-to-End Encryption
ML:	Machine Learning
API:	Application Programming Interface

Table 1: List of Acronyms

OVERVIEW

Introduction:

In an age characterized by rapid technological advancement, the security, privacy, and integrity of data have become critical concerns. Traditional cryptographic techniques have served as the backbone of data protection, but the emergence of sophisticated cyber threats requires a reevaluation of existing security measures. This research thesis explores an innovative approach to data security and privacy through the implementation of geo-enhanced encryption techniques. The primary goal is to strengthen the security of various systems, including laptops and mobile devices, by incorporating user-specific attributes, Pseudo Random Number Generator (PRNG), and a concept called Toleration Distance (TD) into the Advanced Encryption Standard (AES) algorithm.

Chapter 1: Background and Context

This section provides a thorough summary of data security and privacy conditions, highlighting the limitations of traditional cryptographic methods. It discusses the evolution of cyber threats, emphasizing the need for advanced security measures that can adapt to emerging challenges. Additionally, it introduces the AES algorithm, Pseudo Random Number Generator (PRNG), and TD as fundamental components of the proposed geo-enhanced encryption technique.

Chapter 2: Literature Review

The second chapter examines the majority of research on data security, encryption techniques, and geo-enhanced security measures. It discusses the strengths and weaknesses of various cryptographic algorithms, highlighting the need for an approach that integrates geographical and contextual information. This chapter also analyzes previous research efforts in the field of geo-enhanced encryption.

Chapter 3: Methodology

This chapter delves into the methodology employed in the research. It explains the process of augmenting the AES algorithm with user-specific attributes, Pseudo Random Number Generator (PRNG), and TD to create a robust geo-enhanced encryption technique. It also outlines the tools and technologies used for experimentation and data collection.

Chapter 4: Results and Discussion

This chapter presents the results of the experiments and offers an in-depth discussion of their implications. It assesses the strengths and weaknesses of the geo-enhanced encryption technique, comparing it to traditional cryptographic methods. Additionally, the chapter explores potential real-world applications and the feasibility of widespread adoption.

Chapter 5: Conclusion and Future Directions

The final chapter summarizes the key findings of the research and outlines the contributions made to the field of data security and privacy. It discusses the potential impact of the geo-enhanced encryption technique on enhancing data protection in an evolving threat landscape. Moreover, this chapter suggests avenues for future research and development, including potential refinements and expansions of the proposed approach.

References:

This section lists all the sources, scholarly articles, books, and research papers referenced throughout the thesis to provide credibility and support for the research findings and methodology.

Appendices:

The appendices include supplementary materials such as code snippets, detailed experimental data, and any additional information that enhances the understanding of the research process and results.

CHAPTER 1: INTRODUCTION

1.1 Abstract

In our swiftly evolving digital age, data has transcended its role as a mere commodity to become the lifeblood of modern society, powering industries, facilitating global communication, and driving innovation across various sectors. Yet, the digital transformation that has fueled this information revolution has also given rise to a formidable adversary: cyber threats. The ubiquity of technology, the ever-expanding attack surface, and the increasing sophistication of malicious actors have culminated in a complex cybersecurity landscape characterized by perpetual vigilance.

1.1.1. The Challenge of an Evolving Cyber Threat Landscape

The formidable challenge posed by cyber threats is dynamic and ever-evolving, demanding a comprehensive understanding of the continuously shifting landscape. Cyber attackers exhibit a relentless commitment to refining their tactics, deploying a diverse arsenal of cyber weapons. This includes not only sophisticated malware but also an array of ransomware variants, advanced persistent threats (APTs), and zero-day exploits. [1] The complexity and sophistication of these cyber threats present a multifaceted challenge, as adversaries operate with alarming agility, transcending geographical boundaries, and adapting their strategies on a global scale.

The adaptability of cyber adversaries is a significant factor contributing to the inadequacy of conventional cybersecurity defenses. Traditional defense mechanisms often find themselves outpaced and outmaneuvered by the swift evolution of cyber threats [1]. The expansive and interconnected nature of modern digital ecosystems further exacerbates the challenges, as cyber threats traverse diverse environments, exploiting vulnerabilities and evading detection through sophisticated tactics.

Considering this intricate and dynamic threat landscape, there exists an imperative to reassess and fortify cybersecurity strategies. The contemporary cybersecurity paradigm demands proactive measures that go beyond conventional approaches. A comprehensive understanding of emerging threats, coupled with the development of adaptive defense mechanisms, becomes essential in safeguarding against the diverse and relentless nature of cyber adversaries. [2]. The ongoing struggle between defenders and attackers necessitates a continuous evolution of cybersecurity practices to ensure resilience in the face of the ever-changing cyber threat landscape.

1.1.2. The Inherent Limitations of Traditional Cryptography

Traditional cryptographic techniques have long stood as venerable bastions, providing a reliable line of defense against unauthorized access and data breaches. The bedrock of data security, cryptographic algorithms such as the renowned Advanced Encryption Standard (AES), have played an indispensable role in fortifying the integrity and confidentiality of confidential information [2]. The robustness of these cryptographic workhorses has been particularly evident in their ability to withstand brute force attacks, a testament to their efficacy in preserving data security [2].

Nevertheless, the world of cyber threats is dynamic and diverse; they are always changing to take leverage of new flaws and get around established defenses. Despite the formidable resilience of cryptographic algorithms against brute force attacks, their efficacy encounters limitations when faced with the subtle intricacies and sophistication of emerging cyber threats.

The nuanced challenges presented by modern cyber threats go beyond the conventional parameters that traditional cryptographic techniques were designed to address. Sophisticated attack vectors, including advanced persistent threats (APTs), zero-day exploits, and polymorphic malware, pose formidable challenges that extend beyond the capabilities of established

cryptographic methodologies [1]. The adaptability and agility of cyber adversaries necessitate a reevaluation of traditional cryptographic approaches to ensure they remain effective in the face of evolving threats.

In light of these considerations, there arises a critical need to explore and implement advanced cryptographic strategies that can not only withstand the evolving threat landscape but also offer resilience against emerging attack vectors [1]. The journey towards securing sensitive data in the digital realm requires a continuous evolution of cryptographic techniques to address the ever-expanding scope and sophistication of cyber threats.

1.1.3. The Imperative of Adaptive Security

Within a landscape characterized by continual technological innovation and the ever-shifting tactics of malicious actors, relying on static security measures proves inadequate. The imperative for an adaptive security paradigm becomes increasingly evident, calling for defenses equipped to dynamically respond to the emergence of novel threats. Although the traditional cryptographic approach is robust, its static nature underscores the pressing requirement for adaptive security strategies capable of evolving in tandem with the dynamic threat landscape [1].

1.1.4. Contextual Information Integration for Enhanced Security

To reinforce data security and privacy, the strategic integration of contextual information emerges as a promising avenue. User-specific attributes, encompassing factors such as geographical location, timestamp data, and other pertinent contextual information, hold the potential to augment the existing security framework. By infusing cryptographic processes with contextual data, a more resilient and adaptive security posture can be established, aligning security measures closely with the specific circumstances of each data transaction. [2].

1.1.5. Geo-Enhanced Encryption as a Solution

This research seeks to address these multifaceted challenges by exploring and developing geo-enhanced encryption techniques. These techniques, anchored in the integration of geographical and contextual data, represent a promising avenue for bolstering cryptographic resilience [2]. By enhancing established cryptographic algorithms, such as AES, with geo-contextual data, this research aims to create a more robust defense mechanism, capable of adapting to the constantly evolving threat landscape.

1.2 Research Objectives

This study aims to achieve the following main goals:

- 1.2.1 To develop a geo-enhanced encryption technique that incorporates user-specific attributes, Pseudo Random Number Generator (PRNG), and the concept of Toleration Distance (TD) into the AES algorithm.
- 1.2.2 To evaluate the suggested geo-enhanced encryption technique's security and efficacy via thorough research and trial and error.
- 1.2.3 To evaluate the potential real-world applications and implications of the geo-enhanced encryption technique in enhancing data security, privacy, and integrity.
- 1.2.4 To contribute to the ongoing discourse on adaptive security measures that can effectively counter evolving cyber threats.

1.3 Thesis Approach

This thesis' main goal is to create a cutting-edge geo-enhanced encryption technique, a novel approach designed to bolster data security, privacy, and integrity. This technique aims to transcend the limitations of conventional cryptographic methods by incorporating user-specific attributes, time constraints, and the innovative concept of Toleration Distance (TD) into the well-

established Advanced Encryption Standard (AES) algorithm. Through meticulous development, the thesis endeavors to create a robust and adaptive encryption methodology that aligns with the dynamic nature of the contemporary threat landscape.

Following the development phase, the next goal is to thoroughly evaluate the security and efficacy of the suggested geo-enhanced encryption method. This involves comprehensive experimentation and analysis, evaluating the encryption technique's resilience against a spectrum of potential cyber threats. The assessment aims to provide empirical evidence of the technique's efficacy, contributing valuable insights to the field of cybersecurity and adaptive encryption strategies.

Beyond technical validation, the thesis extends its focus to evaluating the potential real-world applications and implications of the geo-enhanced encryption technique. This involves a nuanced exploration of how the developed technique can be practically deployed to enhance data security, privacy, and integrity in various domains. The objective is to bridge the gap between theoretical advancements and practical implementations, offering a holistic understanding of the geo-enhanced encryption technique's applicability in diverse contexts.

Furthermore, the thesis aspires to make a significant contribution to the ongoing discourse on adaptive security measures. In light of the ever-evolving cyber threat landscape, the importance of adaptive strategies cannot be overstated. By addressing this imperative, the thesis aims to contribute valuable insights and recommendations to the broader field of cybersecurity, fostering discussions on how adaptive security measures can effectively counter emerging cyber threats. This multifaceted approach seeks not only to advance the theoretical foundations of encryption techniques but also to provide practical solutions that resonate with the contemporary challenges posed by malicious actors in the digital realm.

1.4 Scope of Research

This research primarily focuses on the development and evaluation of a geo-enhanced encryption technique within the context of the AES algorithm. The study will assess the practicality and effectiveness of this approach in enhancing data security and privacy. While the research acknowledges the importance of addressing other aspects of cybersecurity, such as intrusion detection and network monitoring, these topics are outside the scope of this thesis.

The scope of this research encompasses the comprehensive development, assessment, and real-world evaluation of a geo-enhanced encryption technique designed to augment data security, privacy, and integrity. The primary focus is on integrating user-specific attributes, Pseudo Random Number Generator (PRNG), and the Toleration Distance (TD) concept into the Advanced Encryption Standard (AES) algorithm. The scope extends across the following key areas:

1.4.1 Algorithmic Development

The research delves into the intricacies of enhancing the AES algorithm by incorporating user-specific attributes, Pseudo Random Number Generator (PRNG), and TD. This involves formulating a novel geo-enhanced encryption technique that adapts to the dynamic nature of the contemporary threat landscape.

1.4.2 Experimental Assessment

Rigorous experimentation and analysis form a crucial aspect of the research scope. The assessment aims to empirically validate the effectiveness and security of the developed geo-enhanced encryption technique. A diverse range of cyber threat scenarios will be simulated to evaluate the resilience of the technique under varying conditions.

1.4.3 Real-World Applications

The research explores the practical applications and implications of the geo-enhanced encryption technique across different domains [3]. This involves examining how the developed technique can be implemented to enhance data security, privacy, and integrity in real-world scenarios, considering diverse environments and use cases.

1.4.4 Interdisciplinary Contributions

Contributing to the ongoing discourse on adaptive security measures is a key aspect of the research scope. The thesis seeks to close the knowledge gap between theoretical developments and real-world applications by providing recommendations and insights that are in line with cybersecurity's multidisciplinary character.

1.4.5 Broader Implications

The scope extends to understanding the broader implications of the research findings. This involves considering the societal, ethical, and strategic dimensions of deploying adaptive security measures in response to evolving cyber threats.

By encompassing these key areas, the research aims to contribute a holistic understanding of the geo-enhanced encryption technique, from its algorithmic foundations to its real-world applicability and implications [4]. The scope extends beyond theoretical advancements to address the pressing need for adaptive security measures in contemporary cybersecurity practices.

1.5 Thesis Challenge

The primary challenge in this thesis lies in developing a geo-enhanced encryption technique that not only integrates user-specific attributes, Pseudo Random Number Generator (PRNG), and Toleration Distance (TD) into the AES algorithm but also maintains a delicate balance between enhanced security and practical usability. The complexity arises from the need to

navigate the intricate interplay of these elements within the encryption framework while ensuring that the proposed technique remains adaptable to the ever-changing landscape of cyber threats [5].

The thesis faces the challenge of conducting thorough experimentation and analysis to assess the effectiveness and security of the developed geo-enhanced encryption technique. This involves creating diverse scenarios to simulate potential cyber threats and evaluating the technique's resilience under varying conditions. Ensuring the robustness of the encryption technique against a spectrum of attack vectors is a demanding aspect of this challenge.

Furthermore, evaluating the real-world applications and implications of the geo-enhanced encryption technique introduces another layer of complexity. The challenge here is to contextualize the theoretical advancements within practical scenarios, considering the diverse environments and use cases where enhanced data security, privacy, and integrity are paramount. This entails exploring the feasibility of implementation across different sectors and understanding the potential limitations and constraints associated with real-world deployment.

Contributing to the discourse on adaptive security measures poses its own set of challenges. The thesis must synthesize theoretical insights with practical considerations to provide meaningful recommendations for enhancing cybersecurity in the face of evolving cyber threats. Navigating the interdisciplinary nature of this discourse and effectively communicating the implications of adaptive security measures to a diverse audience is a nuanced challenge that requires a comprehensive understanding of both technical and strategic aspects.

In essence, the thesis confronts the challenge of seamlessly integrating theoretical advancements, empirical validation, and practical relevance to contribute to the evolving field of cybersecurity. Meeting this challenge requires a multidimensional approach, combining technical

expertise, experimental rigor, and a keen awareness of the broader implications of the geo-enhanced encryption technique in the contemporary digital landscape.

1.6 Thesis Organization

The remaining sections of this thesis are structured as follows:

- Chapter 2 provides a comprehensive review of existing literature related to data security, encryption techniques, and geo-enhanced security measures.
- Chapter 3 details the overview of the methodology employed in the research, including the process of augmenting the AES algorithm.
- Chapter 4 presents the solutions of the proposed geo-enhanced encryption technique with the Machine Learning Algorithms to prevent cyber threats.
- Chapter 5 offers a conclusion and outlines future research directions.

By addressing these objectives and delving into the intricacies of geo-enhanced encryption, in light of the constantly changing nature of cyber dangers, our research aims to make a substantial contribution to the fields of data security and privacy.

CHAPTER 2: LITERATURE REVIEW

In an era where data security is paramount, the evolution of cryptographic techniques has become a critical necessity. While traditional cryptographic methods such as the Advanced Encryption Standard (AES), Data Encryption Standard (DES), and the Rivest-Shamir-Adleman (RSA) algorithm have long been the backbone of data protection, the ever-advancing threat landscape necessitates new and innovative approaches. [1] Location-based encryption represents an advancement in technology that expands on the security of traditional cryptographic techniques, offering an approach that can adapt to the challenges posed by an increasingly sophisticated digital environment. [2].

2.1 The Role of Location-Based Encryption

Location-based encryption fundamentally enhances the security provided by traditional cryptographic techniques. It does so by integrating the geographical and contextual data of users into the encryption and decryption processes. This integration can encompass either symmetric or asymmetric cryptographic approaches or even a hybrid combination of both. The utilization of location-based data introduces an additional dimension of security, enabling cryptographic systems to better align with the specific circumstances of each data transaction [1, 2].

The implementation of location-based encryption relies on specific protocols designed to seamlessly integrate geographical and contextual information into cryptographic operations. This chapter delves into key protocols central to the effective utilization of location-based encryption.

2.2 Geo-Encryption Algorithm

The Geo-Encryption Algorithm, introduced by Logan Scott and Dorothy E. Denning in 2003, stands as a significant milestone in the domain of data security and encryption [1]. It is crucial to remember that conventional encryption technologies are the foundation of this creative

strategy and communication protocols, providing a bridge between established cryptographic methods and the dynamic requirements of the contemporary digital landscape.

The fundamental concept underlying the Geo-Encryption Algorithm revolves around encrypting data based on the anticipated position, velocity, and time (PVT) of the intended receiver. This anticipated PVT data serves as the cornerstone for the generation of the geo-lock key. The geo-lock key, in turn, undergoes a bitwise exclusive-OR (XOR) operation with a randomly generated key, culminating in the creation of the geo-lock session key [5]. The geo-lock session key serves as the linchpin of data security, ensuring that the encrypted data remains confidential throughout its transmission.

The transmission of the geo-lock session key to the intended receiver is a pivotal step in the process. Asymmetric encryption is employed to securely transfer the session key, adding an extra layer of security to prevent unauthorized interception. Upon receipt of the encrypted session key, the receiver employs an anti-spoof GPS device to obtain the PVT data. This information serves as a critical component in the generation of the final session key, a key that mirrors the one generated by the sender through a similar process.

However, the Geo-Encryption Algorithm is not without its unique challenges. A central element in this encryption approach is the PVT-to-geo-lock mapping function, which is pivotal for ensuring the successful decryption of the data. The mapping function serves as a bridge between the sender and receiver, and it becomes imperative that both parties possess the same mapping function. This requirement can pose challenges, particularly when the sender and receiver only occasionally communicate [1, 5].

In essence, the Geo-Encryption Algorithm represents an innovative marriage of traditional encryption principles with contemporary security needs. By anticipating the receiver's position,

velocity, and time, and mapping these parameters into a secure key exchange, it provides a robust security mechanism. Nevertheless, the shared mapping function requirement underscores the need for a deeper examination of this approach in real-world scenarios, particularly in cases where sporadic communication occurs. As technology continues to advance, so does the need to balance security and practicality, making innovative approaches like the Geo-Encryption Algorithm a pivotal subject of study in the ongoing quest for robust data protection.

2.3 The Foundations of Geo-Encryption

The core principle underpinning the Geo-Encryption Algorithm is the integration of location-specific data into the encryption and decryption processes [5]. This location data encompasses a diverse array of attributes, including geographic coordinates, time-based information, and other contextual factors. By intertwining these elements into the cryptographic framework, the algorithm fortifies data security and adaptability in the face of contemporary threats.

2.3.1 Geographic Coordinates: The Cornerstone

At the heart of the Geo-Encryption Algorithm lies the utilization of geographic coordinates. These coordinates, often derived from GPS data, serve as the foundational element of location-based encryption. By weaving these coordinates into the encryption process, the algorithm aligns data security with physical locations. This approach mitigates the risk of unauthorized access from remote or unfamiliar locations, as decryption becomes contingent on the verification of specific geographic parameters.

2.3.2 Time-Based Data: A Dynamic Layer of Security

Time-based data plays a pivotal role in enhancing the security provisions of the Geo-Encryption Algorithm. In an environment where cyber threats operate on tight schedules, this

dynamic element introduces a temporal dimension to data security. The algorithm can be configured to limit access to certain timeframes, further reducing the window of vulnerability and bolstering the adaptive nature of data protection.

2.3.3 Adaptability in the Face of Modern Threats

One of the standout features of the Geo-Encryption Algorithm is its adaptability. [5] This adaptability is a response to the ever-evolving threat landscape, where attackers frequently employ novel strategies to breach security systems. By incorporating geographic and contextual information, the algorithm can dynamically adjust its security parameters to align with changing circumstances.

2.3.4 The Broader Implications

The Geo-Encryption Algorithm is not just a theoretical concept; its real-world applications have the potential to transform data security across various sectors. From safeguarding financial transactions to enhancing the privacy of location-based services, this algorithm holds the promise of becoming a cornerstone in the protection of critical information.

2.4 Location Dependent Encryption Algorithm (LDEA)

The Location Dependent Encryption Algorithm (LDEA), initially proposed by Liao et al. [4], presents a noteworthy deviation from traditional encryption methods, offering a unique solution for secure data transmission within mobile information systems. LDEA's design foregoes the mapping function utilized in the preceding Geo-encryption algorithm protocol [5], thus warranting a comprehensive examination of its distinctive approach.

The foundation of LDEA's design centers on the application of a static location-dependent data encryption strategy, wherein latitude and longitude coordinates are intricately incorporated

into the encryption process. By doing so, LDEA not only protects the data during transmission but also imposes constraints on the location where the data can be decrypted.

To ensure the integrity of this system, the Toleration Distance (TD) protocol is introduced to address two primary concerns. First, it resolves the potential inaccuracies inherent in static location data, acknowledging that pinpointing an exact location can be challenging in practice. Second, it mitigates the inconsistency often observed in GPS device receivers.

When a sender transmits the target coordinates and the associated TD, LDEA generates an encryption key specific to the transmission. Upon receiving the encrypted data, the receiver attempts to match its acquired coordinates with the target coordinates within the defined TD range. Successful coordination results in the decryption of the ciphertext back into its original plaintext.

However, it is essential to note a practical challenge associated with the LDEA protocol. Its dependence on static location data necessitates that the receiver decrypts the ciphertext in the exact location where the target coordinates were initially established. In practice, achieving this level of precision with GPS coordinates can be exceptionally difficult, given the inherent inaccuracies associated with GPS positioning technology.

2.5 Dynamic Toleration Distance (DTD)

The Dynamic Toleration Distance (DTD) protocol, introduced by Hamad and Elkour [7], emerges as a progressive response to the intrinsic challenges associated with GPS receivers, primarily addressing the issues of inaccuracy and inconsistency. This protocol substantially enhances its practicality and fortifies its resilience against potential security threats. The essential premise underlying the DTD protocol is the transformation of mobile node location into a dynamic and formidable defense mechanism.

In stark contrast to static encryption methods, which rely solely on the precise position of mobile nodes and a fixed Toleration Distance (TD), the DTD protocol leverages the dynamic nature of mobile node locations. The mobile receiver actively registers a set of coordinates that not only encapsulate their current position but also factor in the velocity during movement. This comprehensive dataset empowers the protocol to estimate the anticipated next location. This calculated position is then used to apply an evolving secret key in conjunction with the DTD.

The incorporation of velocity and movement type as parameters in the DTD protocol amplifies its security profile. By embracing the dynamic nature of mobile nodes' locations, the protocol introduces a level of unpredictability and adaptability that enhances its resilience against attacks. This adaptability aligns with the real-world intricacies of mobile device mobility, providing a more robust defense mechanism compared to static encryption methods.

2.5.1 Transforming Mobile Node Security

The Dynamic Toleration Distance (DTD) protocol represents a pioneering approach to address the challenges posed by the inherent inaccuracies and inconsistencies of GPS receivers within mobile information systems. Introduced by Hamad and Elkour, [7] this protocol ventures beyond traditional static encryption methods by embracing the dynamic nature of mobile node locations and incorporating key parameters to enhance data security.

2.5.2 Dynamic Security Through Mobile Node Movement

The fundamental premise of the DTD protocol hinges on the dynamic location of mobile nodes. Unlike static encryption, which solely relies on a specific position and fixed Toleration Distance (TD), DTD adapts to the ever-changing locations of mobile devices. To accomplish this, the mobile receiver actively registers a comprehensive set of coordinates that encapsulate not only

the present position but also the velocity during movement. This dataset effectively captures the trajectory of the mobile node, laying the foundation for a dynamic security strategy.

2.5.3 Estimating the Next Position

One of the distinctive features of DTD is its ability to estimate the forthcoming location of the mobile node. This estimation is founded on the registered set of coordinates and the velocity data. By extrapolating the anticipated next position, the protocol transforms this into a dynamic secret key. This key evolves as the mobile node moves, continuously adapting to the changing location.

2.5.4 Integration of Movement Type

To further enhance security, the DTD protocol considers the type of movement exhibited by the mobile node. Different types of movement, such as linear, angular, or irregular patterns, can impact the security of the data transmission. DTD accommodates these variations, enabling the protocol to distinguish between intended movements and potentially malicious alterations in mobile node behavior.

2.5.5 Resilience Against Attacks

The dynamic nature of DTD significantly augments its resilience against security threats. By utilizing dynamic location data, mobile node velocity, and movement type as key parameters, the protocol introduces an element of unpredictability and adaptability into the security framework. This adaptability aligns closely with the practical considerations of mobile device mobility, making it substantially more challenging for potential attackers to breach the security of data transmission.

2.5.6 Practical Implications

The practical implications of the DTD protocol are profound. It offers a real-world solution to the accuracy and consistency challenges encountered in mobile information systems. As mobile devices continue to play an increasingly pivotal role in various sectors, from transportation to healthcare, the DTD protocol's adaptability and robust security measures make it an invaluable asset in safeguarding sensitive data during transmission.

2.6 Use Cases

Within the framework of this master's thesis, the primary emphasis is directed towards the examination of three distinct use cases within Location-Based Services (LBS). These encompass services that leverage users' location information for specific functionalities, namely: location-based messaging services, location-based gaming services, and location-based advertisement services. It is crucial to highlight that the protocol under consideration has been meticulously crafted to secure highly sensitive information. The intricacies and demands of these location-based systems align seamlessly with the design principles and objectives of our proposed protocol. Through a focused exploration of these use cases, this thesis endeavors to contribute valuable insights into the efficacy and applicability of the designed protocol in safeguarding sensitive information within diverse LBS scenarios.

2.6.1 Location-based Messaging:

Within the context of this thesis, we delve into the realm of contemporary instant messaging applications that introduce a novel facet to communication by enabling users to send geographically locked messages. In these applications, a user can leave a message tied to a specific geographic area, prompting the recipient to physically visit the designated location for message retrieval. However, a critical concern arises as many of these applications neglect to incorporate

robust measures for safeguarding users' location privacy. In their pursuit of practicality, these applications often capture, store, and utilize location information indiscriminately, resulting in a compromise of users' location privacy.

The proposed solution revolves around the integration of a location-based encryption protocol into these services, offering a paradigm shift in balancing functionality and privacy. Illustrated in Figure 1 is an exemplary scenario depicting the operation of our location-based encryption protocol. In this model, the sender strategically selects a geographic area for message encryption, utilizing the latitude and longitude coordinates as inputs for the location-based encryption protocol. The encrypted message is then transmitted using End-to-end Encryption (E2EE), a measure ensuring that the text remains tamper-resistant and exclusively decryptable at the recipient's end.

Upon receipt of the encrypted message, the recipient initiates a search for the precise location, leveraging location data captured by their device. Only upon arriving at the correct location does the decryption process unfold, allowing the recipient to access the contents of the message. In essence, the location-based encryption protocol stands in stark contrast to conventional location-based messaging applications that operate in a highly location-pervasive manner. The protocol acts as a protective barrier, shielding users from unintentionally divulging their location privacy while still affording them the same functionality present in other systems that disregard the confidentiality of location information.

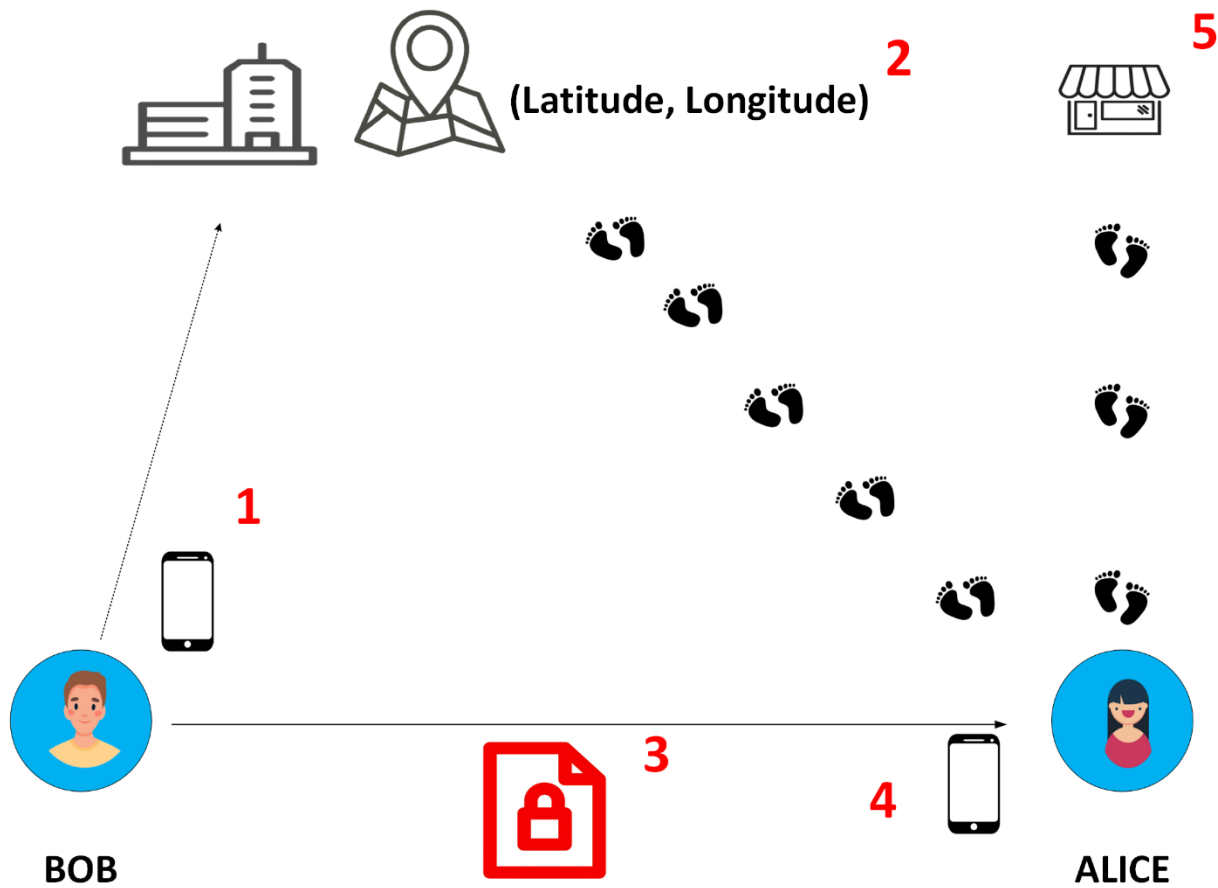


Figure 1: Location-Based Messaging

Sender Bob initiates the process by utilizing his smartphone (1) to acquire the precise location coordinates of a cafe (2), where he intends to secure a message through the location-based encryption protocol. Employing the latitude and longitude of the designated cafe (2), he encrypts the message. Subsequently, leveraging the secure channel established between Bob and Alice, he dispatches the encrypted text (3) to Alice.

Upon receipt, Alice engages her smartphone (4) and traverses different locations such as (5) to decipher the encrypted text (3). Successful decryption hinges on her arrival at the correct location, namely the designated cafe (2). If she aligns with the predetermined location, she can effectively decrypt the encrypted text (3) and access the concealed message.

2.6.2 Location-based Gaming:

The proliferation of devices and the exponential growth in their capabilities have transformed the realm of gaming, turning advanced gaming into a tangible reality in today's world. Notably, many modern mobile devices have evolved into portable gaming tools, expanding the gaming landscape [9]. The allure of gaming lies in its ability to captivate users with challenges, fun, and various elements that evoke a sense of ambition [9]. Prominent examples of this evolution are evident in location-based gaming services, exemplified by games like Pokemon Go and Ingress.

In the context of location-based gaming, users often find themselves consenting to share confidential data, particularly location information, to access these entertainment services. [10]. Unfortunately, this consent, while necessary for gameplay, opens the door for game providers to effortlessly store and utilize this sensitive data. Consequently, everyday locations, including work and home addresses, become readily accessible to game providers, a scenario that challenges the expectation of safeguarding such confidential information [10].

In response to this challenge, the integration of location-based encryption emerges as a pivotal component within gaming frameworks. This encryption methodology introduces a layer of security and privacy by encrypting various in-game elements such as trophies, prizes, or bonus items with specific locations. Users are then incentivized to physically visit these locations to unlock their encrypted rewards, introducing triggering factors such as fun and rewarding feelings.

The application of a location-based encryption protocol in gaming serves a dual purpose. Firstly, it safeguards players from inadvertently divulging their precise location to external entities, addressing the privacy concerns associated with location-based services. Secondly, it enhances the

gaming experience by infusing an element of real-world interaction, encouraging users to explore and engage with their physical surroundings in pursuit of in-game rewards.

In essence, the utilization of a location-based encryption protocol within gaming scenarios not only bolsters privacy protection but also elevates the gaming experience by integrating the real world into the virtual realm [12]. This innovative approach empowers users to enjoy the thrill of gaming without compromising their location privacy, fostering a secure and immersive gaming environment.

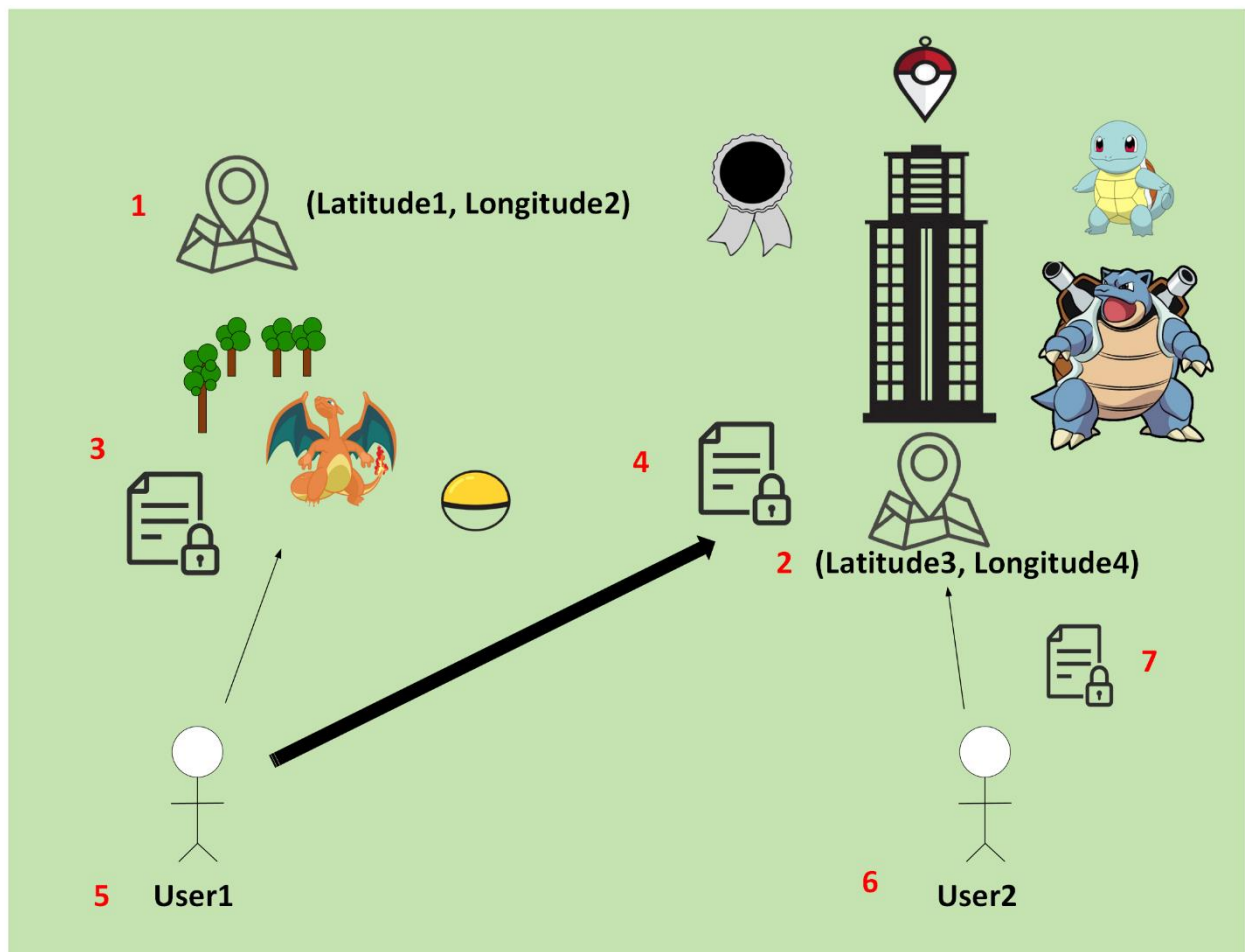


Figure 2: Location-Based Gaming

In this illustrative scenario, User 1 (5) engages in a location-based gaming context, receiving a cipher-text (3) encrypted with the coordinates of a specific park (1), accompanied by in-game

rewards sent by another user through End-to-End Encryption (E2EE). To claim the prize, User 1 (5) is required to physically visit the designated park (1) for decryption. Conversely, User 1 (5) can reciprocate by encrypting bonus items for User 2 (6), utilizing the location of another designated area (2) within the game environment. This process results in the creation of an encrypted text (4) containing game-related rewards and achievements, which User 1 (5) transmits to User 2 (6) through E2EE. Subsequently, User 2 (6) is prompted to journey to the specified location (2) to decrypt the received encrypted text (7) transmitted by User 1 (5).

2.6.3 Location-Based Advertisement Services:

In the shopping industry, [14] companies frequently leverage Location-Based Services (LBS) to enhance customer engagement. An illustrative example involves the deployment of Bluetooth beacons and analogous solutions to dispatch notifications to individuals nearby. However, it is noteworthy that, in numerous instances, the location information of customers may be susceptible to dissemination, as corporations predominantly prioritize utilitarian considerations.

A pertinent application of LBS in the shopping sector involves optimizing sales strategies [15]. For instance, a retail establishment aspiring to augment its sales figures can further entice customers by implementing a methodology wherein discount coupons are encrypted using the specific location of the shop. Subsequently, these encrypted coupons are distributed to both online and in-store customers. This strategic integration of location-based encryption not only enhances the allure of promotional offerings but also underscores the commitment to customer privacy. By employing such innovative measures, businesses can achieve marketing objectives while maintaining a conscientious approach to the security and confidentiality of customer location data [16].

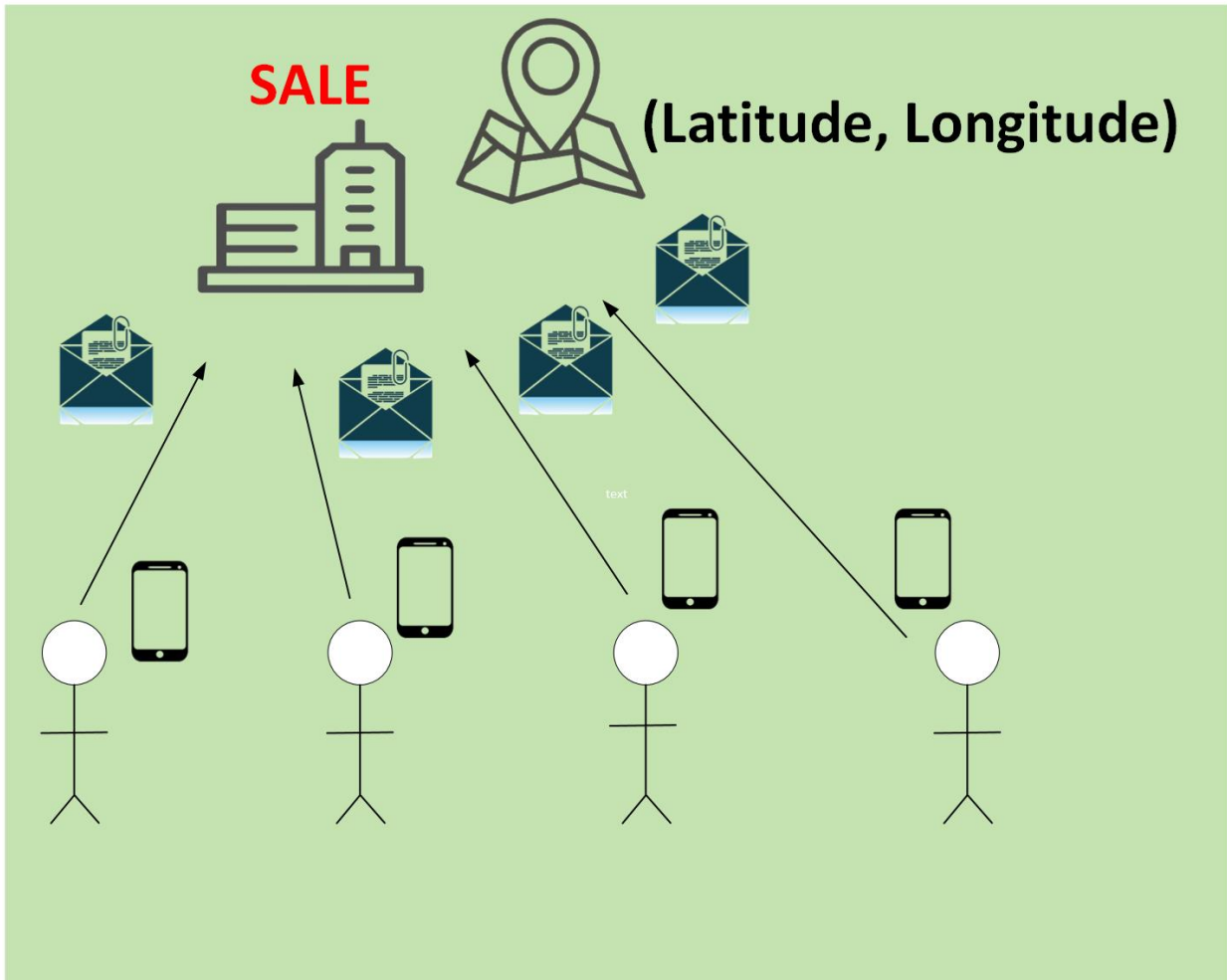


Figure 3: Location-Based Advertisement

In this practical scenario, a local retail establishment generates numerous discount coupons tailored for its clientele. These coupons undergo encryption, incorporating the geographical coordinates of the shop, and are subsequently dispatched to customers' mobile devices utilizing End-to-End Encryption (E2EE). The unique feature of this process mandates that customers physically visit the shop to decrypt the encrypted discount coupons. Upon successful decryption, customers gain access to exclusive discounts applicable for both in-store and online transactions. This innovative approach not only incentivizes customer engagement but also seamlessly integrates the physical shopping experience with digital benefits.

CHAPTER 3: LOCATION-BASED ENCRYPTION

This chapter centers on our meticulously crafted solution, beginning with a comprehensive elevated impression and an in-depth explanation of the protocol. It progresses to a detailed exploration of the functions and intricacies of both the receiver and sender roles, elucidating the operational dynamics of the protocol within the context of these distinct user roles. Following this elucidation, the chapter delves into a comprehensive depiction of our security model, a robust framework justifying the inclusion of specific security countermeasures. Furthermore, it provides a detailed exposition of the threat model, wherein potential threats are meticulously defined. This section specifically focuses on showcasing the structural vulnerabilities of our protocol, offering a thorough analysis of potential weaknesses that might be exploited. The objective of this chapter is to provide a holistic understanding of our designed solution, emphasizing its functionality, security measures, and the delineation of possible threats that have been meticulously considered in its development.

3.1. Overview and Model Portrayal

3.1.1. Overview

This thesis endeavors to establish a secure communication channel that restricts the geographical reception of a message while safeguarding the recipient's location privacy. To elucidate this concept, let's consider a simple analogy involving two key entities, Bob, and Alice, depicting a scenario where Bob, a security enthusiast, seeks to transmit some data to Alice at a specific location. The analogy unfolds as follows:

Objective Setting: Bob aims to transmit data exclusively to Alice at a particular location, like her office location for instance.

Geographical Restriction: Bob ensures that Alice can only access and decipher the message when physically present at the office location, maintaining a strict geographical confinement for message decryption.

Privacy Concerns: Mindful of preserving location privacy, Bob, as a security enthusiast, is intent on preventing any inadvertent disclosure of location to external entities, such as applications or web services.

Application of Encryption Protocol: Bob uses the office location as a reference point to encrypt the message using a location-based encryption protocol before sending it to Alice.

Recipient Verification: Upon receiving the encrypted message, Alice actively engages in determining the accurate location by capturing her location data through her mobile device or other means.

Access Control: If Alice finds herself anywhere other than the designated office location, she remains unable to decrypt the message, as it is specifically encrypted concerning the office location.

Recognition and Relocation: Acknowledging that the message is confined to the office, Alice then proceeds to the office location.

Message Decryption: Upon reaching the office, Alice gains access and decrypts the data sent by Bob, completing the secure communication cycle.

This analogical depiction showcases the secure transmission and reception process designed to restrict geographical access to the message, reinforcing the recipient's privacy by necessitating physical presence at the designated location for decryption. The application of the location-based encryption protocol not only ensures the geographical confinement of the message

but also underscores the importance of maintaining location privacy in secure communication exchanges.

The analogy presented underscores pivotal considerations integral to the protocol: the secure transmission of messages between users, safeguarding location privacy, and confining information within a specific geographical area. However, the crux of this system hinges on the incorporation of location information into the encryption process, a fundamental element in achieving geographic confinement of the message.

The main idea centers on encrypting messages utilizing a symmetric encryption algorithm, incorporating location information as an input to generate a corresponding symmetric encryption key. This methodology entails deriving an encryption key that directly correlates to the specified location. As a result, for a recipient to successfully decrypt the message, they must possess and utilize the exact symmetric key associated with the specific location. Achieving this congruence necessitates the recipient to accurately identify and possess the identical coordinates used in generating the encryption key.

This approach ensures that geographic constraints are applied to information through the deployment of cryptographic components. By intricately weaving location data into the encryption process, the protocol effectively constrains message decryption to specific geographical coordinates. This strategic fusion of location-based elements with cryptographic principles serves as the linchpin in establishing and maintaining geographical limitations on information, underscoring the essence of our research in creating a secure and location-constrained messaging system.

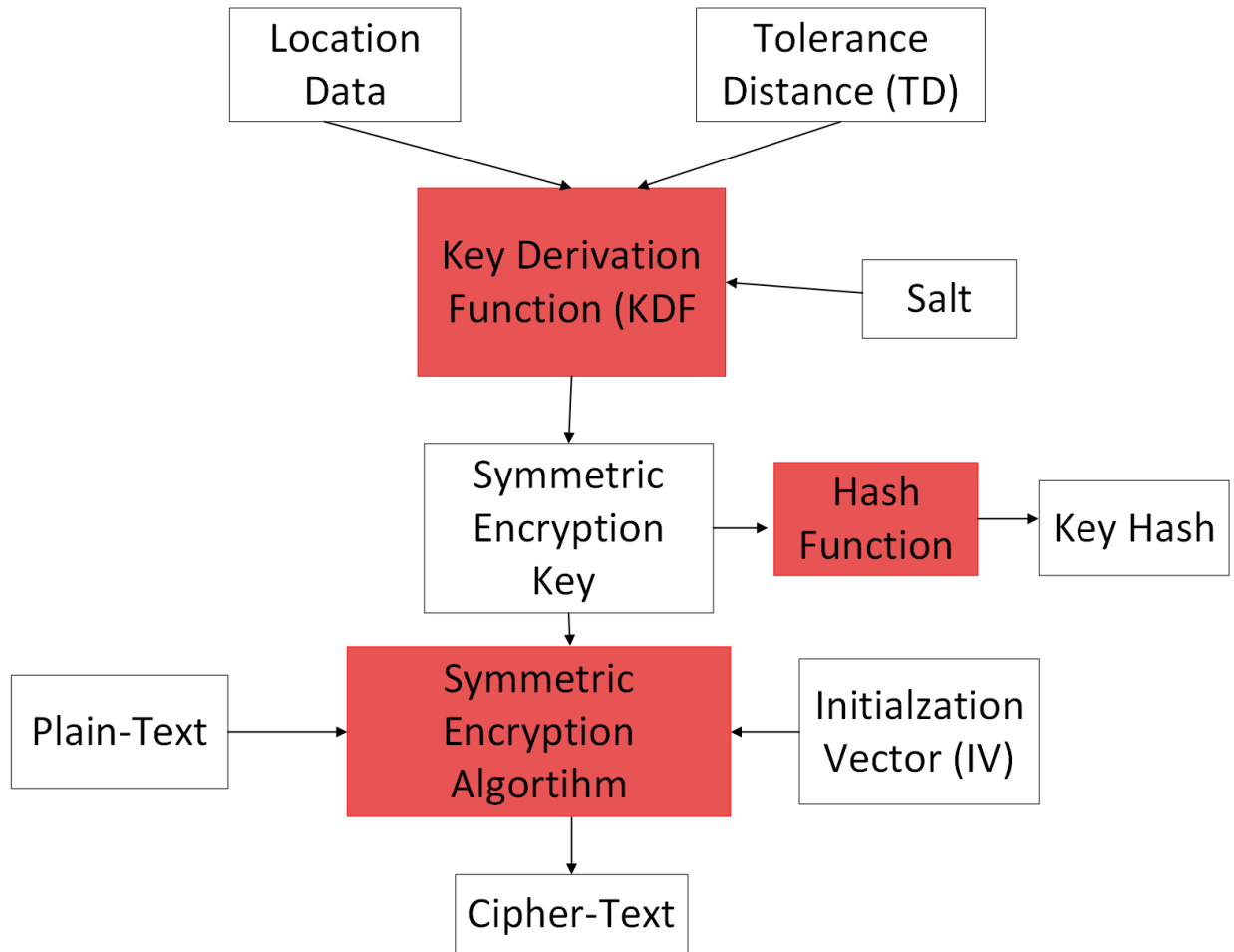


Figure 4: Block Diagram of the Proposed Model

The process of encrypting a given plaintext is delineated in the diagram. The red boxes signify the cryptographic building blocks employed for encrypting the plaintext. Key inputs to the key derivation function include Location Data, Tolerance Distance (TD), and Salt. Following key derivation, a hash function is employed to compute the hash of the key. Subsequently, the derived key serves as an input to a symmetric encryption algorithm, along with the plaintext and initialization vector (IV), facilitating the encryption of the plaintext through the symmetric encryption algorithm.

It is imperative to note that encrypting a message with location data results in the geographical confinement of the message to a specific position [19]. Consequently, the protocol must be designed to incorporate a degree of tolerance in the decryption range. Failure to adhere to this condition implies that even a minor displacement of the recipient's precise position concerning the actual message position would lead to unsuccessful decryption. Figure 6 provides a comprehensive overview of the protocol, emphasizing the derivation of the symmetric encryption key from location data for message encryption.

3.1.2. Protocol Description

For a more comprehensive and systematic interpretation, let's deconstruct Figure 4 by analyzing its components from top to bottom. This process enables us to gain valuable insights into each step as we navigate through the elements. As previously mentioned, our approach employs a symmetric encryption algorithm, a mechanism where both the sender and recipient mutually establish a single shared confidential element, referred to as an encryption key. With this key and the original message (referred to as plain text), encryption generates an unintelligible and random sequence (known as cipher text), effectively securing the message. Notably, this process is executed while incorporating location data for enhanced security.

Stepping back to examine symmetric encryption keys, it becomes evident that there are essentially two primary methods for creating a cryptographically secure encryption key. One approach is to employ a Cryptographically Secure Pseudorandom Number Generator (CSPNG), which functions as a Pseudo Random Number Generator (PRNG) with tailored characteristics designed for cryptographic applications. Another method entails leveraging a Key Derivation Function (KDF), which is a procedure for deriving a cryptographically robust key from pre-existing confidential data, such as another key or an initial secret, such as a password.

In the context of this thesis, our chosen method for deriving the encryption key involves employing a Key Derivation Function (KDF). This decision is underpinned by the availability of initial keying material, specifically our location data. By utilizing a Key Derivation Function, we can derive a robust and secure encryption key leveraging the inherent security of our location data, thus reinforcing the overall security of the encryption process within the scope of this research.

Various password-based Key Derivation Functions (KDFs), extensively detailed in section 3.3.1, utilize a preliminary secret, such as passwords, as the basis for deriving encryption keys. Prominent examples include (PBKDF2) Password-Based Key Derivation Function 2, Bcrypt, and Scrypt. Notably, Bcrypt and Scrypt are intentionally engineered to be memory-intensive, requiring significant and adjustable memory usage to execute computations efficiently. This deliberate emphasis on memory intensity serves to impede the speed of a brute-force attack, where the invader/attacker systematically tests every possible key to uncover the encryption key. In contrast, PBKDF2 demonstrates lower memory overhead, a critical consideration, especially for mobile devices aiming to mitigate substantial battery drainage across diverse scenarios. Consequently, for our specific requirements, PBKDF2 is selected as the preferred Key Derivation Function.

As illustrated in Figure 4, the Key Derivation Function (KDF) takes three inputs as parameters: Location Data, Tolerance Distance (TD), and Salt. Location Data comprises the geographical coordinates of the individual, including lat/long values (Latitude and Longitude). Tolerance Distance (TD) represents the distance, measured in meters, that extends the decryption range, thus accommodating error acceptance. Conversely, Salt enables the reuse of a master key for generating multiple derived keys by employing a unique Salt for each use. This approach thwarts potential attacks, like rainbow table attacks, as the unique Salt makes it unfeasible to derive the same key solely from the initial secret, latitude, and longitude information. The uniqueness of

Salt values ensures that even if the input key materials remain constant in subsequent iterations within the Key Derivation Function, the resulting encryption key diverges due to the distinctive Salt values.

The uniqueness of Salt becomes pivotal in determining the desired number of different keys for derivation. Typically, 32 and 64-bit Salt values are preferred, both of which offer extensive variation. To cater to the need for diverse keys, 32-bit Salts are utilized due to their resilience to exhaustion. Furthermore, the Key Derivation Function is specified with 1,000,000 iterations, strategically designed to deliberate this process, striking a balance between efficiency and enhanced security by slowing down the operation without compromising overall performance.

Finally, our choice for the symmetric encryption algorithm is the Advanced Encryption Standard (AES), selected for its widely acknowledged robustness, rendering it practically impervious to breaches. AES operates in counter-mode (CTR) with a 128-bit specification, primarily due to its capacity for reduced memory consumption, a critical aspect beneficial for mobile devices, leading to lower battery usage.

With the symmetric encryption key derived from the Key Derivation Function, as delineated in Figure 4, our messages are primed for encryption using this algorithm. However, returning to the analogy involving Bob and Alice, an essential query emerges: How can Alice ascertain the correctness of the key derived from a potential location, preventing a misleading outcome from an incorrect key? To address this quandary, the sender takes a hash of the encryption key and transmits it to the receiver alongside the ciphertext. This facilitates the receiver's key derivation process, allowing a reference for comparison when deriving keys during the search for the correct location.

Consequently, during the receiver's quest for the accurate location, keys are generated based on diverse locations. By hashing these keys, the receiver can juxtapose them against the received key hash from the sender. A match in the key hashes validates the decryption process. To achieve this, Secure Hash Algorithms (SHA)-256 are employed to obtain the key hash and for the key derivation function, primarily due to the compromised integrity of SHA-1. In the conclusive step, the message undergoes encryption via a symmetric encryption algorithm, represented in Figure 6, incorporating three essential inputs: plaintext, encryption key, and Initialization Vector (IV). The plaintext signifies the message to be encrypted, the encryption key serves as the symmetric key for encryption, and the initialization vector is instrumental in averting data repetition, thereby thwarting attempts to uncover patterns and decipher the ciphertext through dictionary attacks. A 16-byte IV size is adapted to align with the block size of the encryption algorithm, optimizing its effectiveness.

3.1.3. Geographic information utilization during the key derivation process.

Now that we've gained an understanding of how the protocol functions. Let's delve into the detailed procedure of generating the encryption key using tolerance distance and location data. Figure 5 offers a visual representation of the derivation of a symmetric encryption key from lat/long values, in combination with the TD (tolerance distance). To begin, let us explore the nature of the data received from location. The receiver actively captures their location through a mobile device, often represented in the format of Degrees Decimal Minutes. For instance:

- N 24°24.8900' indicates 24 degrees 41.3501 minutes North

- E 9°67.0871' signifies 9 degrees 67.0871 minutes East

This Degrees Decimal Minutes representation allows latitude values within the range of 0 to 89 degrees and longitude values within 0-179 degrees. When interpreting these values, the point

sign acts as a reference. The initial digits before the point sign indicate the degrees component of the location data, while the subsequent numbers, located to the left of the point sign, represent minutes, allowing for floating-point precision. As minutes reach a maximum of 59, these initial digits convey the minutes along with the decimal part.

In the subsequent phase, we introduce our Tolerance Distance (TD) in meters to complement the location data. Through a process of multiplying the latitude and longitude values by 10,000 and subsequently dividing the resulting value by the specified tolerance distance, we form distinct quadrants based on the tolerance distance parameter. It's important to note that a 1-meter alteration in latitude and longitude corresponds to approximately 5.4 and 69 units respectively.

This method delineates quadrants delineated by the tolerance distance, effectively grouping multiple location points that fall within the same quadrant. Consequently, decryption access to the information is granted to entities within the same quadrant, obviating the necessity for pinpoint accuracy in decrypting at an exact location. The rationale behind the initial multiplication of location data by 10,000 relates to precision. As depicted in Figure 5, latitude and longitude values accommodate up to four digits after the decimal point. Hence, when these values are divided by the tolerance distance and multiplied by a value, let's say 1,000 instead of 10,000, the resultant value would lack precision due to the truncation of the remaining digits, rendering it less accurate. This discrepancy would result in a coarser addressing of the designated location point, reducing the precision of the decryption process.

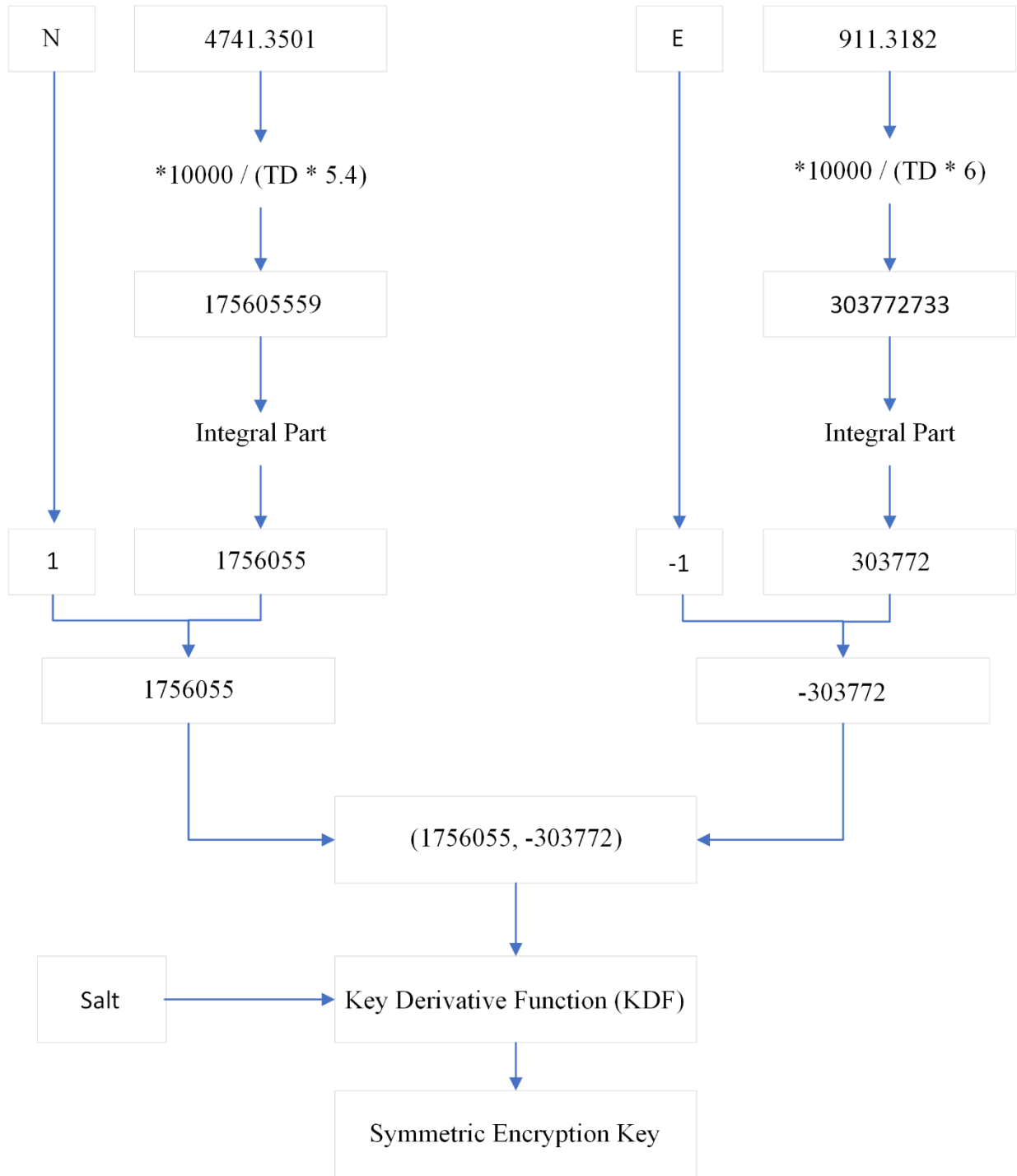


Figure 5: Utilization of Geo Location in KDF

Lat/Long (Latitude & Longitude) data are integrated with the Tolerance Distance to formulate a quadrant, enabling additional location points within the tolerance distance to generate the same input for the key derivation function. Subsequently, this input, combined with the Salt,

is utilized in the key derivation function to generate the symmetric encryption key. In this illustrative example, a tolerance distance of 5 meters is assumed, and the corresponding inputs are computed.

To elucidate, consider Figure 5 for the calculations. Let's assume a reference point at N 4741.3514. Additionally, suppose two location points, A and B, with relative distances of 4 and 15 meters to the reference point, respectively, have location values of N 4741.3536 and N 4741.3595. These calculations are made within a tolerance distance of 5 meters. Now, let's explore the implications of multiplying the location data by 1000 instead of 10000.

To illustrate the implications of these calculations, consider Figure 5 for reference. Let's take a reference point, N 47°41.3514, and introduce two additional location points, labeled A and B, positioned at a relative distance of 4 and 15 meters, respectively, from the reference point. Location A is at N 47°41.3536, while location B is positioned at N 47°41.3595, with a specified tolerance distance of 5 meters.

Let's simulate the steps indicated in Figure 5 for both locations, applying the multiplication by 1,000 instead of 10,000:

1. For Location A: $47^{\circ}41.3536 * 1,000 = 47,413,536$
2. Location A: $47,413,536 / (5 * 5.4) = 1,75,605.68$
3. Location A: Extracting the integer part: 1,75,605
4. For Location B: $47^{\circ}41.3595 * 1,000 = 47,413,595$
5. Location B: $47,413,595 / (5 * 5.4) = 1,75,605.907$
6. Location B: Extracting the integer part: 1,75,605

This example demonstrates that although the two location points are positioned at 4 and 15 meters from the reference, they yield the same resultant integer value when the location data is

multiplied by 1,000, although the tolerance distance is set at 5 meters. As a result, for both points A and B, the side to the left of the input string for the key derivation function, shown in Figure 5, would produce the same value. Point A satisfies this requirement, even though Point B is outside the 5-meter tolerance limit. However, to perform the same steps with a multiplication of location data by 10,000, the resulting values would be 1,75,6056 and 1,75,6059 for A and B, respectively. This scenario exhibits a more precise outcome from a location accuracy perspective, showing higher precision and accuracy in the results by maintaining sensitivity instead of rounding off the numbers.

Upon factoring in the Tolerance Distance (TD) and isolating the integer part of the location data, we now incorporate the location sign. Latitude values are typically designated as South (S) and North (N), while longitude values are denoted as East (E) and West (W). For the Western and Northern hemispheres, the location sign is set to 1, while for the Eastern and Southern hemispheres, it is designated as -1, providing crucial hemisphere information to the data. These manipulations based on tolerance distance result in the multiplication of the output with the location sign to append hemisphere information to this data.

Upon completing these computations, both latitude and longitude values, along with the assigned salt, are employed as inputs for the key derivation function, facilitating the generation of the symmetric key.

3.2. Detailed Examination of Sender and Receiver Perspectives

This section builds upon the prior outline of the protocol and the process of generating a symmetric key using location data. It delves into the operational dynamics between the sender and receiver within the protocol.

Figure 8 delineates the communication flow between a sender and receiver. Initially, the sender determines two critical values: the location data representing the message's intended location for encryption and the TD (tolerance distance) in meters, which defines the decryption range. These parameters serve as inputs for the KDF (key derivation function), alongside the Salt. Once the key is derived, the sender uses a hash function to obtain a key hash derived from the key. This key hash serves as a reference for the receiver when attempting to decrypt the message by generating their key. Subsequently, an initialization vector (IV) is employed, taking the encryption key as an input for the symmetric encryption algorithm. The plaintext undergoes encryption and is transmitted via an end-to-end encrypted channel.

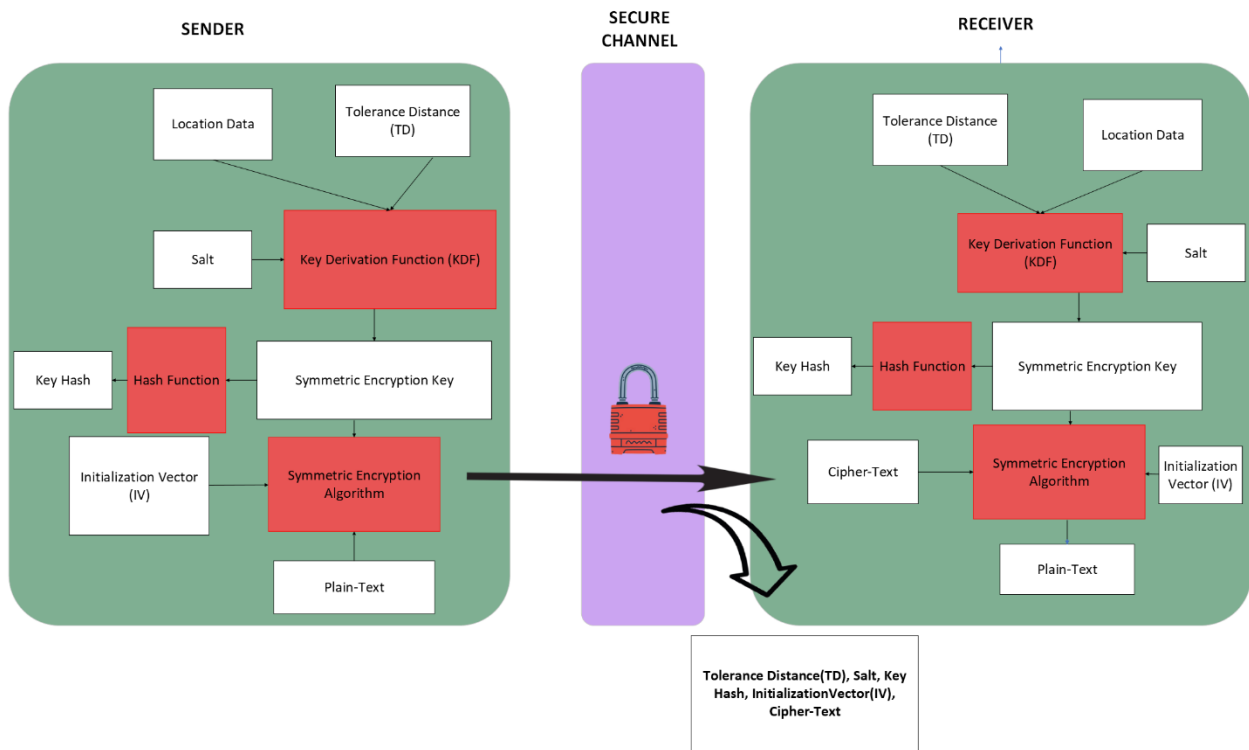


Figure 6: Overview of the Proposed Model

To derive a consistent encryption key, several essential components, including Tolerance Distance (TD), Salt, Key Hash, Initialization Vector (IV), and the cipher-text, are transmitted to the recipient through the secure communication channel facilitated by End-to-End Encryption

(E2EE). Armed with this information, the recipient possesses all the necessary elements to derive the identical key. Significantly, the inclusion of the key hash allows the recipient to verify the accuracy of the generated decryption key, ensuring the integrity of the decryption process.

Naturally, for the receiver to derive the identical key, essential factors need to align between the sender and receiver. This includes mirroring the Salt and (TD) Tolerance Distance parameters for the key derivation function. Subsequently, the receiver must obtain a reference key hash directly from the sender, enabling a comparison with their own derived key. Lastly, to decrypt the message successfully, the receiver must possess an identical initialization vector (IV) in addition to the key and the cipher text. This underscores the necessity for the receiver to access the Tolerance Distance (TD), salt, key hash, initialization vector (IV), and ciphertext, all crucial components enabling the generation of an identical cryptographic key. Consequently, the sender securely transmits all this indispensable information through the designated secure channel to ensure the receiver can derive the same cryptographic key.

3.3. Security Requirements

3.3.1 Computational Expensiveness

As the potential for a receiver to act maliciously remains a concern, it's crucial to scrutinize the utilization of location data within the key derivation function. Within the current protocol version, location data serve as inputs for the key derivation function. However, there is a mechanism to verify the location claim of an entity. This opens avenues for exploitation, allowing receivers to potentially subvert the protocol by submitting fabricated location data through various deceptive means, including the manipulation of Global Positioning System (GPS) information or even through brute-force tactics. The integration of location information into the key derivation

function is significant. Yet, it's important to note that cryptographic key derivation functions serve multiple purposes, contingent upon their specific applications:

Key Split-up: Key derivation functions play a pivotal role in generating multiple keys derived from a primary cryptographically secure encryption key. This facet of key derivation finds application in scenarios where distinct keys are necessary for individual communications, such as an authentication server interacting with numerous clients, each requiring unique keys for their exchanges. Additionally, for heightened security, a server might mandate the use of a new encryption key for each communication session. Utilizing a unique Salt per session allows for the derivation of multiple keys, demonstrating the versatility of key separation.

Key Expansion: Key derivation functions aren't solely restricted to generating multiple keys; they can also modify the existing key configuration into a predefined format. In specific instances, a 128-bit encryption key might be transformed into a 256-bit key, based on the belief that the latter provides enhanced security under specific conditions.

Key Stretching: This aspect of key derivation functions is tailored to tackle low-entropy initial passphrases, like a user password with limited entropy, say around 30 bits. The intention is to use such passphrases for encryption and decryption while impeding unauthorized access via brute-force attempts. This deliberate slowing down of the process through a high iteration number serves to fortify the encryption against guesswork or brute-force attacks on the passphrase.

Notably, the deliberate deceleration of the key derivation function through a high iteration count serves as a defense against brute-force attacks, significantly elongating the key generation process. By employing a computationally intensive key derivation function, the time taken for key generation is immensely extended, dissuading malevolent receivers from inundating the system with fabricated location values.

3.3.2 Nonce Information

In the previous segment, a range of key derivation functions was examined, with particular emphasis on leveraging key-stretching within these functions. As previously mentioned, these functions incorporate multiple parameters serving as supplementary data to bolster security and enhance functionality. Our approach specifically harnesses the use of Salt to counter rainbow-table attacks, a technique where an attacker employs precomputed tables comprising an extensive array of strings usable as inputs for key derivation functions. Subsequently, the attacker compares the hash function outputs to seek a match, aiming to backtrack and identify the corresponding input.

The subsequent discussion in section 3.4.2 delves into the underlying issue associated with location data. It highlights the limited key space resulting from the confined geographical scope of Earth and the concentration of human habitation in specific regions. This limitation renders the location data highly vulnerable to guesswork. Consequently, a malicious user aiming to decrypt the cipher text might employ two strategies. They could conduct a brute-force attack, attempting every feasible input within the key derivation function to decrypt the cipher text, or they might opt for a rainbow-table attack. The brute-force approach is notably expensive, while the latter is considerably more accessible for an attacker to execute. In response to these challenges, the use of Salt serves as a pivotal measure to counter pre-calculated tables when assailants attempt to guess the limited location data. Given that the Salt remains undisclosed to the attacker, they would need to generate a new table for each unique Salt, rendering the process as resource-intensive as a brute-force attack. Therefore, the incorporation of Salt as an input into the key derivation function effectively mitigates the susceptibility to rainbow-table attacks.

When employing the Advanced Encryption Standard (AES) to encrypt plain text, the resulting ciphertext's susceptibility to attacks and information exposure largely depends on the

chosen AES mode. For instance, in the Electronic Code Book (ECB) mode, utilizing the same plain text and encryption key consistently generates identical ciphertext. However, in the Counter (CTR) mode, the inclusion of an Initialization Vector (IV) enables the creation of diverse ciphertexts even when encrypting the same message with the same key, provided distinct IVs are used per operation. This variability in IV usage across multiple encryptions contributes to the generation of different ciphertexts. In effect, an attacker lacking this specific information would not be able to execute such an attack on the data.

3.3.3 "Utilizing End-to-End Encryption (E2EE) for Data Security"

To prevent unauthorized data dissemination by external entities, detailed in section 4.2.1 distinguishing between insider and outsider attackers, our protocol employs End-to-End Encryption (E2EE). This method ensures a secure channel for communication between the sender and receiver, restricting data readability exclusively to these endpoints. Achieving this involves encrypting a message using asymmetric-key encryption, known as public-key cryptography. E2EE operates with two correlated keys: public and private. Each user safeguards their private key while sharing their public key with other users. A sender encrypts a message using the receiver's public key, rendering the only decryption path through the corresponding private key held by the intended receiver. This process effectively inhibits intermediaries, as they lack access to the requisite private key, thereby safeguarding against tampering, as elucidated in the prior section outlining adversaries.

Given the inherent lack of entropy in location data, as demonstrated in the preceding chapters, we combine this data with Salt to avert rainbow-table attacks. Should an external attacker obtain the Salt, they could easily execute a rainbow-table attack, considerably diminishing the key space available due to the limited entropy of location data. Additionally, the acquisition of

components like key hash and ciphertext could provide the attacker with exploitable information under specific conditions, enabling assaults such as cipher-text-only attacks. Considering these vulnerabilities, the utilization of E2EE significantly mitigates these issues.

3.4. Thread Model

In this segment, we delve into the threat model, revealing the structural vulnerabilities within our location-based encryption protocol.

3.4.1 Adversaries

In this section, we define two potential adversaries and outline their capabilities to exploit possible flaws within the protocol. The first adversary, an insider user, poses a threat by attempting to read messages without meeting the requisite conditions, such as being at the specific location designated by the sender for decrypting encrypted text. This user may deploy fraudulent GPS software or manipulate device settings to spoof their location, endeavoring to decrypt messages by providing fake locations.

Conversely, an outsider could seek to intercept and manipulate information within the E2EE-secured communication channel. While unable to directly decrypt the ciphertext due to the lack of the corresponding private key, this adversary could intercept messages and masquerade as a receiver. By encrypting incoming messages using their public key and forwarding them to the actual receiver using the receiver's public key, the attacker could potentially deceive the sender and receiver into believing they are communicating directly, provided they avoid detection during this process.

3.4.2 Key Derivation Function (KDF) Input Space

We utilize lat/lon (latitude and longitude) data to pinpoint an entity's geographic location, serving as inputs for the key derivation function in consort with the Salt. Given that the receiver

possesses the salt during decryption, this knowledge allows the receiver to narrow down the decryption area. A relatively small location dataset can be utilized due to the Earth's geographical reality. It poses a direct threat to the key space within the protocol, as the Salt is already within the receiver's knowledge.

Assessing the likelihood of identifying the correct location for various tolerance distances (10, 20, 50, and 100 meters) concerning the urbanized area of the earth, roughly 5.1×10^{14} square meters, with only 3% urbanized. This assessment involves determining the total count of different possibilities by dividing the Earth's urbanized area by the area of a circle corresponding to the given tolerance distance.

Possible Location Points Based on Tolerance Distance (TD)

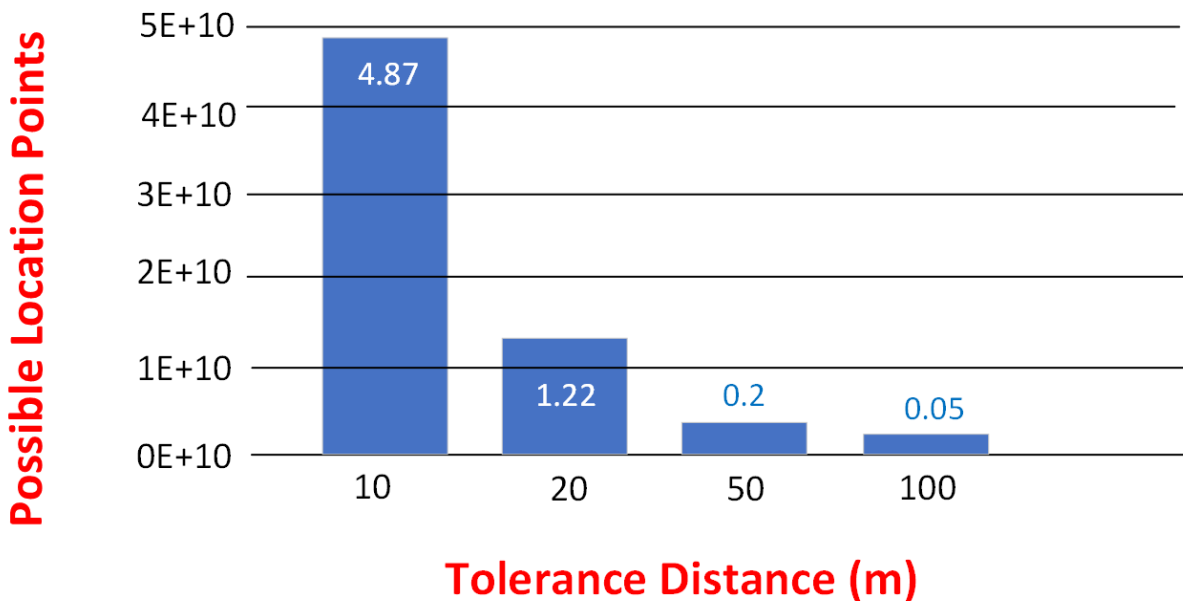


Figure 7: Possible Location Points

The calculation of potential locations within a given tolerance distance relies on Earth's surface area, considering that only a mere 3% is urbanized. This analysis reveals a notable

reduction in the number of feasible locations, owing to the substantial coverage of water sources on Earth. As the tolerance distance is expanded, the impact of Earth's predominant water-covered areas becomes increasingly pronounced, influencing the overall count of viable locations.

For a 10-meter tolerance distance,

$$\frac{1}{5.1 \times 10^{14} \text{m}^2 \times (3.14159 \times 10^2) \text{m}^2} = \frac{1}{4.87 \times 10^{10}}$$

For a 20-meter tolerance distance,

$$\frac{1}{5.1 \times 10^{14} \text{m}^2 \times (3.14159 \times 20^2) \text{m}^2} = \frac{1}{1.22 \times 10^{10}}$$

For a 50-meter tolerance distance,

$$\frac{1}{5.1 \times 10^{14} \text{m}^2 \times (3.14159 \times 50^2) \text{m}^2} = \frac{1}{0.2 \times 10^{10}}$$

For a 100-meter tolerance distance,

$$\frac{1}{5.1 \times 10^{14} \text{m}^2 \times (3.14159 \times 100^2) \text{m}^2} = \frac{1}{0.05 \times 10^{10}}$$

The derived conclusions from the four equations and Figure 7 reveal a direct correlation between tolerance distance and the size of the key space. A larger tolerance distance leads to a significantly reduced key space. The implications are clear: a larger tolerance distance makes it substantially easier for an insider attacker to compromise the key, posing a significant security risk. If location information were processed using a standard hash function with contemporary processor speeds, the vulnerability to key space attacks would be heightened. To counteract this risk, a computationally intensive key derivation function is utilized, effectively deterring a potential receiver from exploiting this relatively diminished key space through brute-force methods.

3.4.3 Confidentiality of Communication Channel

In section 3.3.1, the critical role of Salt was delineated. As established, acquiring the Salt, distinct for each message, significantly diminishes the potential key space for the key derivation function. Consequently, should an outsider attacker gain access to the Salt, they could employ the same brute-force tactic, compromising the security measures employed by the key derivation function. The confidentiality of the Salt, alongside other transmitted data like the IV, ciphertext, and key hash, is fundamentally contingent on the robustness of the E2EE framework. The security lies in ensuring the Salt remains undisclosed, crucially reducing the potential input space for the key derivation function.

In theory, end-to-end encryption endeavors to prevent intermediaries from intercepting or altering messages. Consequently, the 16-byte salt remains concealed from outsiders, effectively eliminating the threat of rainbow attacks due to the prohibitively high computational cost it would incur for an attacker. However, recent research suggests that despite the assurances of E2EE within a service, its security might be subject to interpretation contingent upon the implementation medium such as a mobile application. Nevertheless, for the scope of this master's thesis, the examination of this nuanced aspect is deferred for future research studies.

3.4.4 False positive and false negative values

During the decryption process, success is expected only within the defined tolerance distance. As demonstrated in Figure 5, the integers derived from latitude and longitude data serve as a representative quadrant, forming a grid reflecting the Earth's surface. The analysis reveals that the comprehension of the encryption key for the receiver is based on the same quadrant logic begins to falter as it nears the quadrant boundaries. In instances where the resultant integers are

near these boundaries, attempts to construct the same encryption key fail, leading to erroneous negative values and unsuccessful decryption efforts.

This scenario is exemplified in Figure 5, where the integer component of the fractional number, obtained by dividing the latitude and longitude by the tolerance distance, showcases potential discrepancies. During the decryption attempt, the receiver's provided location data may yield an integer differing within a range of 0 to 1 compared to the value calculated by the sender, even when the receiver's location is well within the defined tolerance distance.

Let's take a latitude of 4741.3503 N and a tolerance distance of 10 meters as an example. This position's resultant integer, 878027, is in contrast to another place 10 meters distant, designated by a latitude value of 4741.3555 N, which yields 878028 as the consequent integer. Despite the latter location being within the defined tolerance distance, the false-negative value prevents the successful decryption of the ciphertext. To address this challenge, the receiver is enabled to assess the left and right adjacent quadrants alongside its quadrant. While this resolves the issue, it introduces a scenario where some receivers can decrypt the information, even if they are outside the defined tolerance distance, leading to false-positive values.

Adjacent quadrants are involved while examining the 4741.3555 N point, where the resultant number after being multiplied by 10,000 and divided by the tolerance distance is 878028. Hence, the resultant integers 878027 and 878029 are also tested as inputs to the key derivation function, providing a total of 9 different combinations of 3 possible values for each quadrant, encompassing both the latitude and longitude dimensions, to be used as inputs for the key derivation function.

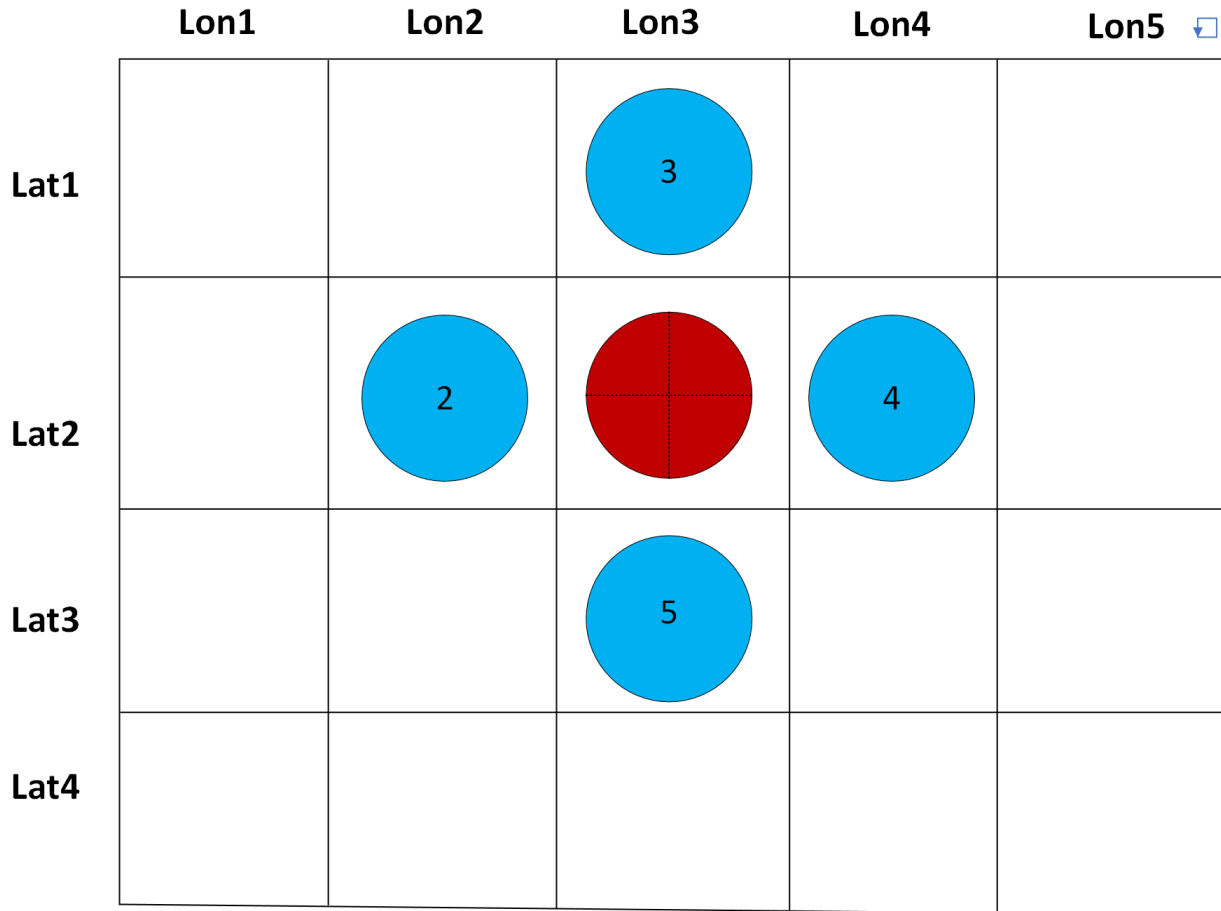


Figure 8: Location points Accuracy

In the provided grid, the identified quadrant for the encryption key is highlighted by the red circle (1). Despite the utilization of Tolerance Distance (TD) to expand the decryption area, the computed quadrant does not comprehensively cover the entire region within its bounds. Proximity to the quadrant boundaries results in the exclusion of certain location points falling outside the tolerated area. Consequently, when the receiver computes the decryption key, a thorough examination of adjacent quadrants becomes imperative. This entails checking the left (2) and right (4) quadrants for latitude, as well as the north (3) and south (5) quadrants for longitude. Consequently, the receiver must explore nine distinct combinations to encompass adjacent quadrants for both latitude and longitude values.

In Figure 8, an example is presented to illustrate the sender's position-based key generation constrained by a tolerance distance. As in the math example, the sender's quadrant falls short of covering all its values, especially near the boundaries. This failure at the border regions within the same quadrant results in the inability to generate the same key, leading to occurrences of false-negative values. On the other hand, the receiver's ability to assess 9 quadrants might allow decryption, even if not in the exact quadrant, producing false-positive outcomes.

3.4.5 Location Verification Claims

Location-based systems (LBSs) and encryption systems reliant on location data encounter significant challenges in verifying location claims. Despite cryptographic security providing robust encryption and meeting security standards, the vulnerability lies in the susceptibility of the end systems to manipulation, such as key theft from user devices. Verification of location claims becomes a critical concern due to the ease with which mobile devices can manipulate their reported location, pretending to be in different places. This vulnerability presents a serious issue when the location claims lack secure verification, specifically within the encryption process. Encryption occurs solely based on location data without authenticating the claimed location, leaving it highly vulnerable to manipulation.

Present solutions attempt to address this by incorporating methods like "location-proofs" formed through collaboration with nearby peers or employing witnessing approaches using external servers for location verification, like triangulation. Nonetheless, it's crucial to highlight that maintaining location privacy is of paramount importance and shouldn't be compromised by sharing this information with unintended recipients. In essence, our protocol is deficient in location verification, emphasizing the need for further research in this area for a more comprehensive solution.

3.5. Security Analysis

The system faces a dual threat from distinct adversaries, illustrated in Figure 6. The first scenario concerns potential interference from external sources aiming to eavesdrop on or disrupt the ongoing transmission. The interception might lead to the suspension of data packets or unauthorized monitoring of the communication flow, potentially breaching security and disrupting the intended exchange of information. Here, the security of the transmission is at risk due to external interference and monitoring.

In the second case, an internal entity specifically, the receiver might adopt a malicious approach to access messages without adhering to the required location parameters. This deceitful endeavor involves fabricating locations using fake GPS signals, effectively deceiving the system by presenting false locations while not physically occupying those spaces. Additionally, an insider located in a more restricted area might leverage their knowledge of common encryption points. By predicting these probable encryption spots, they could potentially exploit these common elements to decrypt the message, sidestepping the necessary location requirements and enabling unauthorized access to the secured information. To mitigate these security threats, the system is fortified with countermeasures designed to prevent unauthorized external interference and thwart internal attackers, ensuring the resilience of the data transfer process.

3.5.1 Location Authentication

In this section, the security of the protocol is scrutinized through the lens of location authentication. In a previously defined context (referred to in Chapter 1), a set of conditions was established to underpin a secure location-based encryption system:

1. The decryption of a message is exclusively permissible when the user is physically present at the designated location.

2. Messages should only be decipherable within a specifically confined geographical area, ensuring that the decryption is location-bound.

3. The system should disallow a user from decrypting a message for a second time without physically being present at the specified location, particularly when receiving a new message at the same location.

These conditions highlight the need for integrating location information into the encryption scheme. The challenge lies in effectively incorporating geographical constraints into the existing cryptographic foundations to restrict decryption access based on location.

The proposed protocol addresses this by employing location data as an authentication element in the key derivation process. This integration is supplemented by the inclusion of a defined range, termed the Tolerance Distance. This distance parameter serves to provide flexibility in decryption areas, acknowledging that a designated location might encompass more than a single point.

The protocol's design includes the creation of quadrants that aggregate multiple positions falling within the same quadrant concerning latitude and longitude. This grouping system, relative to the specified tolerance distance, allows users within the defined geographical range to successfully decrypt messages. Consequently, individuals located outside this range lack the necessary inputs for the key derivation function, effectively preventing unauthorized decryption attempts.

When devising such a scheme, a plethora of challenges emerge, primarily due to the inadequacy of location information for this specific cryptographic task. The deficiency lies in the lack of a crucial cryptographic component entropy. In the realm of cryptography, the complexity of encryption keys is pivotal; it should be computationally infeasible for an attacker to derive

information. Entropy embodies the inherent randomness within data that renders it arduous for an attacker to predict or guess successfully.

However, the issue with location data, as extensively discussed in preceding chapters, stems from the notable coverage of Earth's surface by water sources and the relatively limited urbanized areas. This geographic reality substantially diminishes the randomness property of location information. Furthermore, the tendency for a sender to leave a message within proximity, often known to the sender's acquaintances, further contributes to geographical predictability, rendering location information significantly inadequate for securing sensitive data. The weakened randomness characteristic makes it unsuitable for direct utilization in an encryption scheme requiring robust security.

Utilizing location information directly as an encryption key poses additional challenges, notably the predictability of recurring messages encrypted with the same location data, which invariably generates the same encryption key each time. To circumvent this vulnerability, the protocol employs a unique Salt for each message, thus generating distinct encryption keys for different messages. This strategic use of individualized Salts prevents the repetitive creation of the same key, frustrating decryption attempts that rely on predicting the location.

In conjunction with Salt, leveraging key derivation functions involves increasing the iteration number. This deliberate augmentation renders the decryption process computationally intensive. Consequently, this approach serves to mitigate the problem of low entropy, enhancing the robustness of the encryption scheme. The inherent issues associated with location information, as delineated earlier and to be further explored in subsequent sections concerning data verification, significantly impede the adoption of location-based encryption schemes for highly sensitive systems, such as those employed in the domains of banking and military applications. The

paramount concern in securing such critical systems necessitates rigorous safeguards, and the limitations of location-based encryption impede meeting those stringent security demands.

However, for systems aimed at more specific use cases, as elucidated earlier, this protocol offers a highly efficient protective measure to confine information within designated geographical boundaries. In contexts where the sensitivity of the system permits such restrictions and where the specific functionalities align with the capabilities of the encryption approach, this protocol demonstrates its efficacy in providing robust security measures.

3.5.2 Location Disclosure

In the preceding section, we detailed the security implications of utilizing location data as an encryption key. Using location data directly as an encryption key could expose vulnerabilities, potentially allowing attackers to launch brute force attacks or leverage precomputed tables due to the limited key space. To counteract these threats, we introduced computational complexity to slow down these intrusive processes significantly. Chapter 1 established the most pivotal attributes of a secure location-based application: preserving the user's location privacy and ensuring it remains undisclosed and unaltered to any external entities or intermediaries. The significance of maintaining this privacy was emphasized in the introductory section due to the potential repercussions of location privacy breaches. The conjunction of a constrained key space with compromised location privacy significantly threatens the confidentiality of conversations. Consequently, to avoid any security issues, any information about location data or auxiliary parameters used in encryption needs to be kept private.

To address this concern, our primary countermeasure involves implementing End-to-End Encryption (E2EE), a protocol founded on public key cryptography. E2EE establishes a secure communication channel between two users, effectively preventing any external entities from

intercepting or eavesdropping the conversation. This robust protection safeguards against outsider entities who might possess crucial information such as ciphertext, salt, and initialization vectors, which, if exploited, could significantly heighten the attacker's success rate in launching an attack.

However robust End-to-End Encryption (E2EE) might be it doesn't entirely absolve security issues. An attacker's strategy might pivot away from attempting to crack the encryption directly; instead, the attacker could masquerade as the intended recipient, intercept the sender's messages, encrypt them using their public key, and then re-encrypt and send the messages to the true recipient, all without detection a method commonly known as a man-in-the-middle attack. To prevent this attack, methods such as generating unique, one-time character strings based on users' public keys have been proposed [25].

Moreover, regardless of the strength of encryption keys, the vulnerability of users' mobile devices is a key concern. If these devices are compromised, their actual private keys could be stolen, rendering the entire encryption scheme susceptible to exploitation. However, within this scenario, E2EE remains one of the most secure methods for handling data transmission, given its widespread adoption in many of today's popular services for ensuring confidentiality.

3.5.3 Location Verification

In the preceding sections, our focus lay on the security aspects inherent in the use of location data and the privacy measures governing the communication channels. This segment specifically targets the veracity of the location data received, notably from the verification standpoint. While formulating a location-based encryption protocol, the principal query isn't solely about the encryption process using location data, but fundamentally about the robustness of the verification mechanisms for these location inputs.

The validation of location claims assumes paramount significance, as the integrity of any location-based protocol hinges on the authentication of such data for encryption purposes. Yet, vulnerabilities manifest when counterfeit GPS data or similar manipulations distort the perceived location of a device. This exploitation allows malicious users to introduce spurious location data, jeopardizing the reliability of encrypted messages derived from location-based inputs. A multitude of solutions have been proposed, including the utilization of dedicated location-authentication servers or the leveraging of nearby users to act as witnesses, establishing location proofs frequently through Bluetooth. These methods, however, frequently compromise location confidentiality, compelling user location data disclosure for the sake of authentication. Within the current protocol design, an essential shortcoming emerges no measures are in place for authenticating the provided location data. This gaping void paves the way for the utilization of counterfeit location data, posing a serious threat. Authentication of location claims is a multifaceted challenge that demands dedicated research. We defer this complexity to future studies that will delve into the complexities of ensuring robust location verification while ensuring confidentiality is not compromised. Achieving this balance without any form of data dissemination is an intricate realm warranting in-depth exploration.

CHAPTER 4: PREVENTING GPS/GEO-LOCATION SPOOFING IN ANDROID APPLICATIONS

In recent times, the proliferation of various types of location-based service (LBS) applications has generated significant interest among both users and service providers. In these applications, the user's location serves as a pivotal element. Location-based services (LBSs) harness this information to offer a variety of services, including locating nearby friends, identifying local social events, and delivering real-time updates on traffic conditions. The accuracy of users' claimed locations is paramount for the effective functioning of these services.

Users stand to benefit from accurately reporting their locations in various scenarios. Consider an LBS application providing discounts to users who frequent a particular store; precise location reporting is crucial to avoid mistakenly sending coupons to ineligible users. Likewise, in healthcare, a doctor falsely claiming to be in a specific hospital ward to access patient records underscores the misuse potential of location information. Furthermore, Social network applications that help users find nearby friends rely on accurate location sharing to maintain their relevance and utility. In essence, the reliability of location data is pivotal for the integrity and functionality of a wide array of location-based services.

Regrettably, certain applications have emerged intending to assist users in falsifying their location information. These applications are designed to safeguard users' location privacy while they engage with online platforms. However, this poses a challenge as many location-based services (LBS) applications inherently rely on the accuracy of users' real-time locations. In response to the threat of users providing false locations to LBS applications, various central and dispersed location-proof programs have been proposed. These schemes are designed to enable service providers to authenticate users' locations effectively.

In a formal context, location proof refers to an electronic certificate that substantiates an individual's presence at a specific geographic location during a specific timeframe. The primary challenge lies in devising a location-proof scheme that upholds users' privacy during the collection of location proofs. Furthermore, such a scheme must possess the capability to identify users attempting to deceive the system by submitting false location proofs for the locations where they are not physically present. Lastly, the design must be well-suited for smartphone platforms/OS, considering their constraints in processing power, memory, and wireless bandwidth. Balancing these considerations is crucial to the successful implementation of an effective and privacy-respecting location-proof scheme.

4.1 ANDROID METHODS

This chapter explores the practical implementation of the location-based encryption protocol discussed throughout this thesis. The development process involved meticulous consideration of various cryptographic and security aspects. The results obtained from the implementation provide valuable insights into the efficacy of the protocol in ensuring location-based security and privacy.

Numerous methodologies are available to counteract users attempting to manipulate location data on mobile devices. Developers can employ a combination of various location-tracking techniques to identify potential adversaries engaging in location spoofing. Several prevalent technologies in geographic tracking include:

GPS Reporting: Global Positioning System (GPS) [30] technology furnishes the place and time information of devices equipped with a GPS receiver. Utilizing multiple satellites, GPS requires substantial power to receive signals accurately. All satellites transmit on a uniform

frequency, employing Code Division Multiple Access (CDMA) for signal encoding. Spoofing GPS devices can occur through the transmission of deceptive signals resembling authentic GPS signals or by replicating signals generated in a different location at a different time.

A meticulously secured and encrypted Global Positioning System (GPS) incorporates sophisticated measures such as selective availability (SA) and anti-spoofing mechanisms, thereby transforming the GPS infrastructure into a temporally and spatially unpredictable system. This advanced variant of GPS guarantees assured Time, Position, and Navigation, capabilities while boasting a high level of resistance against jamming and cyber threats. The inclusion of complementary sensors within the phone, such as Inertial Measurement Units (IMUs), assumes a pivotal role in fortifying PNT and discerning the accurate location beyond the conventional GPS. These IMUs feature precision-engineered accelerometers and gyroscopes, enabling them to track the phone's movement without relying on external references.

The primary objective of a secure GPS is to maintain functionality despite vulnerabilities like spoofing and jamming attacks. Jamming involves deploying interference signals to disrupt GPS signal reception, and straightforward countermeasures exist to counteract jamming. The encryption technology known as "M-code" [29] enhances jam resistance. Effectively thwarting spoofing involves tracking the device to pinpoint its exact location before any spoofing attempts commence as a preventative measure that halts the device from falsifying its location.

Through the encryption of GPS signals, users can ensure the reception of authentic and secure signals, resilient against spoofing attempts. Tailoring the protective features, sensor support, and enhanced timing capabilities of secure GPS can be customized based on developers' preferences and the intended application requirements. Future iterations of secure GPS systems can thus incorporate varying levels of protection and functionality.

GSM Reporting, governed by the Global System for Mobile (GSM) [21], establishes a communication standard for cellular networks, facilitating air-encrypted wireless communication between mobile phones and base transceiver stations (BTS) or cell towers. Through triangulation by nearby cell towers, a user's location can be determined at a specific time, allowing for the tracking of user movement. The robustness of this method makes spoofing the location exceptionally challenging, requiring external hardware for any attempted manipulation. Furthermore, the encryption of cell traffic, employing a pre-shared key for user authentication, enhances security in this communication framework.

LAN Reporting involves a local area network (LAN) [22], which is a computer network connecting devices primarily through Wi-Fi. Utilizing Wi-Fi access points, users accessing the Internet can have their location accurately determined. While this method is susceptible to spoofing, incorporating a robust encryption algorithm, as recommended in [23], can effectively prevent such security vulnerabilities.

WAN Reporting involves a wide area network (WAN) [25], which is a telecommunication network designed for long-distance connections. Although this method is susceptible to spoofing, it has found extensive use in mobile communications.

Bluetooth technology, akin to Wi-Fi, employs wireless signals from "Bluetooth tags" [26] in real-time location systems to determine a user's location. However, the limited data transmission range makes it less practical for Location-Based Services (LBS).

The Location Assistant [27] has introduced a "stop mocking location" application comprising the following four steps:

1. Request location updates at fixed intervals with specific accuracy.
2. Seek user permission to access the device location.

3. If the requested location service (e.g., GPS) is inactive, guide the user to enable it.
4. Detect and reject mock locations. If mock locations are detected, prompt the user to disable them.

These steps are crucial for obtaining reliable location updates on the Android platform, with the gray boxes indicating user-dependent decisions. Step 4 specifically aims to discern the authenticity of location information by detecting mock locations, leveraging the Android API. For API levels 18 and higher, the `Location.isFromMockProvider()` function is used to flag mock locations.

Developers can employ these technologies to enhance the resilience of an application against spoofing attempts. For instance, if an application unlocks features based on the user's specific location, a combination of GPS and cell tower checks can be implemented. Currently, GPS spoofing applications cannot mimic cell towers. It is important to note that if a user permits execution as a root, these techniques can be circumvented.

4.2 Unsupervised Machine Learning

In the realm of unsupervised learning, a training set comprises samples without explicit labeling. The primary goal of unsupervised learning is to uncover inherent patterns or partitions within the training set, where the data lacks a designated target attribute [31]. Techniques within unsupervised learning aim to scrutinize the data, discerning underlying relationships among its components.

Clustering, a prominent technique within unsupervised learning, focuses on identifying similarity groups within the data, commonly known as clusters. The algorithm employed in clustering endeavors to discern and categorize data points that exhibit proximity to one another into cohesive groups, contrasting them with data points situated farther apart, which are allocated

to separate clusters. The efficacy of a clustering outcome hinges on factors such as the algorithm employed, the distance function applied, and the specific application context [32].

K-Means Clustering: The k-means algorithm serves to partition data into k clusters, where each cluster is characterized by a cluster center, denoted as a centroid. The user defines the value of k, shaping the number of desired clusters. The stepwise procedure for the k-means algorithm, given a predetermined value of k, unfolds as follows [32]:

Random Initialization: Begin by randomly selecting k data points (seeds) to serve as the initial centroids or cluster centers.

Assignment to Closest Centroid: Associate each data point with the centroid that is nearest to it.

Centroid Recalculation: Recalculate the centroids based on the current cluster memberships.

Convergence Check: Verify if a convergence criterion is met; if not, return to step 2

This iterative process continues until a convergence criterion is satisfied, ensuring that the clusters stabilize based on their centroids.

The k-means algorithm stands out as a widely embraced and efficient clustering method, primarily owing to its simplicity [33]. Despite its popularity, it is essential to acknowledge that, like other clustering algorithms, k-means possess certain limitations. It is noteworthy that the efficacy of clustering algorithms is contingent upon the specific characteristics of the data or the applications in which they are employed. There is no definitive evidence establishing the superiority of one clustering algorithm over another, as performance outcomes are inherently linked to the peculiarities of the given scenario. Consequently, the most accurate measure of

clustering algorithm performance lies in empirical testing with real-world data, allowing for a comprehensive comparison of results.

In the context of this thesis, the selection of the k-means algorithm was motivated by several factors. Firstly, its open-source implementation [34] within the scikit-learn framework provided a practical and accessible means of implementation. Additionally, k-means is a renowned and extensively utilized algorithm in the domain of unsupervised learning, further justifying its suitability for the objectives outlined in this study.

4.3 ML Algorithm to Detect Location Spoofing in Android Applications

Within our threat model, we consider the scenario where Alice retains the ability to disable the machine learning application at will or employ a counterfeit location application whenever necessary. However, we posit that Alice will not sustain these actions indefinitely, as there is limited utility for her in consistently providing false locations, especially if she intends to utilize location-based services. Therefore, we assume that Alice will predominantly disclose her authentic location, enabling our machine-learning algorithm to discern her typical behavior. It is essential to note that even in cases where Alice refrains from utilizing the machine learning application or abstains from submitting her genuine location, our proposed protocol maintains the capacity to assess the validity of Alice's location through the utilization of the third phase of the location proof scheme.

4.3.1 Proposed Solution

Our protocol is structured around three distinct phases. The initial phase termed the "Android mock location" phase involves our application scrutinizing the Android platform to ascertain the status of mock location settings. Following this, the second phase, denoted as the "machine learning" phase, entails feeding the user's location history into the machine learning

(ML) algorithm. This algorithm is designed to discern the authenticity of newly submitted locations, and its decision is communicated to the Location-Based Service (LBS) provider. The integration of the machine learning algorithm enhances the confidence of our nearby friends' application in receiving a genuine location, although it does not empower the user to independently verify the authenticity of her submitted location to the LBS service provider.

To address this, our protocol introduces a third phase the "location proof" phase. In this stage, we leverage any existing location-proof scheme to furnish evidence of the submitted location to the LBS service provider. The LBS service provider's ultimate decision regarding the validation of the submitted location hinges on the outcomes derived from both the "machine learning" phase and the "location proof" phase. This multifaceted approach ensures a robust validation process for the submitted location within the context of our protocol.

Unsupervised Learning Phase: Due to uncertainty regarding the authenticity of visited places in our dataset, the utilization of supervised machine learning algorithms is not feasible. Consequently, we opt for an unsupervised machine learning approach, applying a training set devoid of labels. The primary objective of unsupervised learning is to identify natural partitions within the training set, where the data lack a target attribute [35]. This category of learning techniques involves analyzing the data to discern inherent connections. The lifecycle of our predictive analytics protocol is visually depicted in Figure 9, showcasing the progression of this unsupervised machine learning phase.

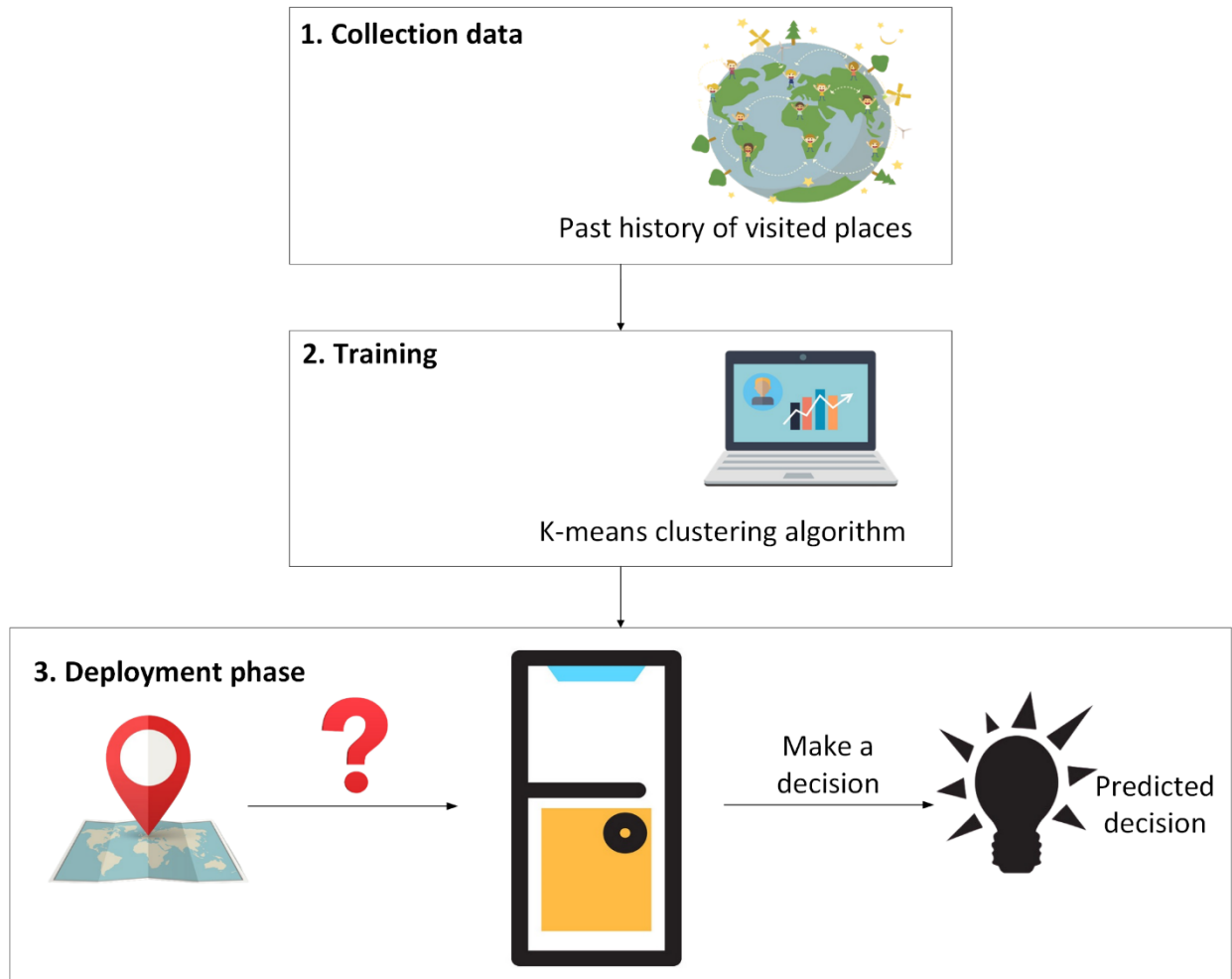


Figure 9: Overview of K-means Algorithm

During the training phase, we employ the k-means clustering algorithm to generate clusters of visited places from the training dataset. The cluster generation involves selecting a specific location as a reference point and defining a radius. All places falling within this radius are included, and the mean of these locations is determined as the new center point. This iterative process continues until the mean ceases to change. Subsequently, all locations within this radius are consolidated into a single cluster, effectively removing them from the dataset. This method aids in creating meaningful clusters representative of visited places in the training data.

After creating clusters from all datasets, each cluster is assigned a unique ID. To identify potential spoofed locations, we consider both the category of visited places and the time gaps in

the dataset, indicating the duration required for the user to transition between two distinct locations. During the prediction step, our algorithm assesses whether a visited place from the testing dataset belongs to one of the generated real location clusters and categories. Additionally, it verifies the presence of a specific time interval "t" between the visited place and the previous one. If these criteria are met, the visited place is labeled as a genuine user's location. Otherwise, it is considered a fake location and subsequently removed from the dataset. This approach enhances the accuracy of distinguishing between authentic and spoofed locations in the testing dataset.

4.3.2 Implementation and Results

The k-means is an unsupervised classification model that is employed on the dataset to discern the authenticity of user-submitted locations, distinguishing between real and fake locations. To evaluate the effectiveness of the proposed protocol, we opted to assess recall, precision, and the Matthews correlation coefficient, alongside accuracy. Relying solely on accuracy may yield deceptive outcomes, especially in cases of imbalanced datasets.

To conduct our analysis, we utilized open-source implementations of the k-means clustering model, leveraging the sci-kit-learn open-source Python library. The datasets were partitioned into training and testing sets to enhance accuracy, with most data allocated for training and a smaller portion for testing. Table 1 presents the outcomes of the implemented methods for the user, considering test dataset sizes of 50%, 40%, 30%, and 20% of the original dataset. This comprehensive evaluation ensures a robust assessment of the protocol's performance under varying conditions.

During our experimentation with the k-means algorithm, a challenge arose concerning the alignment of k-means clustering results with our ground-truth state [36]. The crux of the issue lies in the inherent nature of the k-means algorithm as a clustering, rather than a classification,

algorithm, functioning on unlabeled data. The absence of a predefined correlation between the clusters generated and the ground-truth state of unlabeled data poses a significant challenge. However, in our case, the dataset is known, as the user is the author of this thesis, and can categorize clusters as either "real location" or "fake location."

To address this challenge, we employ the k-means clustering algorithm on the element attributes without incorporating labels. Subsequently, we compare the resultant clusters with the assigned labels, aiming to establish a semantic mapping between our cluster centroids and classes. This mapping, coupled with a measure of "confidence" in the model's performance against the known classes, is derived by evaluating a test set against the centroids. The proximity of a test row to the closest centroid, guided by the semantic mapping, determines the predicted class. Given the labeled nature of the test set, we obtain our ground truth, enabling the construction of vectors for "actual" and "predicted" values, essential for creating the confusion matrix. The "confidence" measure obtained factors in the errors observed in a cluster centroid, influenced by the semantic mapping and ground truth from labeled training rows. It's worth noting that for future testing, especially in scenarios without ground truth and involving multiple users, n-fold cross-validation will be implemented to enhance the robustness of the test results. This rigorous methodology ensures a comprehensive evaluation and validation of the k-means algorithm's performance in diverse scenarios.

As depicted in Table 3, the outcomes illustrate that the algorithm demonstrates accuracy in correctly identifying true positives, primarily due to the larger volume of data associated with our "real location" state. However, challenges arise when the algorithm attempts to ascertain true negatives in certain scenarios, attributable to two main factors. Firstly, the limited quantity of data corresponding to our "fake location" state contributes to this challenge. Secondly, certain instances

of our "real location" states are misclassified into the "fake location" state. This misclassification occurs when the user submits a fake location from her habitual visitation locations, and the time intervals align reasonably with the previous location. Consequently, the algorithm erroneously categorizes the fake location as a real one. These complexities underscore the importance of refining the algorithm further, especially in scenarios with imbalanced data distributions and nuanced user behaviors.

Dataset	True State							
Total Entries	50% for Training and 50% for Testing		60% for Training and 40% for Testing		70% for Training and 30% for Testing		80% for Training and 20% for Testing	
Test Results	Real Location	Fake Location	Real Location	Fake Location	Real Location	Fake Location	Real Location	Fake Location
Real Location	TP = 140	FP = 10	TP = 155	FP = 5	TP = 175	FP = 3	TP = 185	FP = 2
Fake Location	FN = 15	TN = 27	FN = 10	TN = 35	FN = 7	TN = 42	FN = 5	TN = 48
	Recall = 0.9032 MCC = 0.8102 Precision = 0.9333 Accuracy = 0.8931		Recall = 0.9394 MCC = 0.8741 Precision = 0.9688 Accuracy = 0.9221		Recall = 0.9615 MCC = 0.9207 Precision = 0.9836 Accuracy = 0.9464		Recall = 0.9737 MCC = 0.9523 Precision = 0.9894 Accuracy = 0.9594	

Table 2: Summary of the Accuracy of K-means

4.4 Machine learning models for data analysis

For increasing the efficiency and effectiveness of the data it must pass through seven major steps for accuracy and relevancy. Steps include.

4.4.1 Data Preprocessing

4.4.1.1 Preprocessing

It is the process of transformations applied to our data before feeding it to the algorithm. Data Pre-processing is a technique that is used to convert raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for analysis.

4.4.1.2 Splitting of the data set in Training and Validation sets

Train Test Split is one of the important steps in Machine Learning. It is very important because your model needs to be evaluated before it has been deployed. And that evaluation needs to be done on unseen data because when it is deployed, all incoming data is unseen. The main idea behind the train test split is to convert the original data set into 2 parts.

Train: consists of training data and training labels.

Test: consists of testing data and testing labels.

The easiest way to do it is by using scikit-learn, which has a built-in function `train_test_split`.

4.4.1.3 Taking care of Missing values

Missing data is a common problem, and it occurs when a dataset has no value for a feature in an observation. Most machine learning models require data with a value for all features in each observation. In such models, missing data may lead to bias in the estimation of the parameters and compromise the accuracy of the machine learning models.

As a result, we may end up drawing wrong conclusions about data. Therefore, missing data is harmful to machine learning models and requires appropriate handling. There are several techniques we use to handle missing data. They include:

Deleting the observation with the missing value(s)

Mean Imputation

Regression Imputation

4.4.1.4 Taking Care of Categorical Features

We can convert the categorical values into numerical labels. It enhances prediction model accuracy by minimizing noise and non-linearity in the dataset. Finally, binning facilitates detecting outliers, and invalid, and missing numerical data.

4.4.1.5 Normalization of Data Sets

In machine learning, normalization is converting data into the range [37] (or any other range) or simply onto the unit sphere. Normalization and standardization are beneficial to several machine learning methods, especially when Euclidean distance is used. The goal of normalization is to change values to a common scale without distorting the difference between the range of values.

4.4.2 Comparison of Different AI Models

4.4.2.1 KNN Classifier

Nearest neighbor classification is a machine learning method that aims at labeling previously unseen query objects while distinguishing two or more destination classes. Like any classifier, it generally requires some training data with given labels and, thus, is an instance of supervised learning. We have a look at the dataset's features, including 6000 rows \times 37 columns. The analysis looks at how many unique values are present. Then, we must check the shape, count, and data information like type, memory usage, etc. We will extract features to evaluate model performance using the same Features selected for the Nearest Neighbor Classifier, and Naïve Bayes Classifier.

- Training_size = 0.8
- Test_size = 0.2
- Apply KNeighbors Classifier Model KNN accuracy (in %): 85.26%

4.4.2.2 Naive Bayes – Classifier

Naive Bayes is a classification algorithm for binary (two-class) and multiclass classification problems. It is called Naive Bayes or idiot Bayes because the calculations of the probabilities for each class are simplified to make their calculations tractable. In this step (1) Imported Libraries (2) Loaded the dataset and performed (3) Pre-Processing steps:

- Check missing values: No missing values found
- Identify datatype: All data have the same datatypes
- Cleanse data: Removed all duplicate data

Then checked for outliers: Removed outliers by Z_score then performed Feature extraction same as for Nearest Neighbor Classifier, Naïve Bayes Classifier, and Decision Tree. Scaling and then Splitting the dataset into Training and Test Set

- Training_size = 0.8
- Test_size = 0.2
- Apply Naive Bayes- Classifier Model
- Naive Bayes accuracy (in %): 83.58%

4.4.2.3 Comparison of AI Models

AI Model	KNN Classifier	Naive Bayes	K-Means	Decision Tree
Type	Supervised	Supervised	Unsupervised	Unsupervised
Total Entries	80% for Training and 20% for Testing	80% for Training and 20% for Testing	80% for Training and 20% for Testing	80% for Training and 20% for Testing
Result	Accuracy = 0.8526	Accuracy = 0.8358	Accuracy = 0.9594	Accuracy = 0.8894

Table 3: Comparison of AI Models

Given the uncertainty surrounding the reliability of visited places in our dataset, employing supervised machine learning algorithms is deemed impractical. The table further demonstrates the superior accuracy of the unsupervised machine learning K-Means algorithm. Hence, we have chosen to utilize this unsupervised learning model over its supervised counterpart.

CHAPTER 5: CONCLUSION AND FUTURE WORK

5.1 Conclusion

In conclusion, the thesis delves into the intricate realm of location-based encryption systems, exploring the challenges and security considerations inherent in safeguarding information based on geographical data. The primary focus revolves around developing a protocol that integrates machine learning algorithms to assess the validity of a user's claimed location in an Android application.

The thesis systematically addresses the vulnerabilities associated with location-based systems, emphasizing the critical need for robust verification mechanisms in the face of potential manipulations, such as false location claims. It articulates the inherent weaknesses in existing cryptographic models, particularly when confronted with the dynamic and manipulability of location data.

The proposed protocol unfolds in three crucial phases. The initial phase involves scrutinizing the Android platform for mock locations, a fundamental step in ensuring the integrity of subsequent assessments. The second phase introduces a machine learning algorithm, specifically an unsupervised k-means clustering model, designed to scrutinize user location history and discern between genuine and fake locations. The algorithm's performance metrics, including recall, precision, and the Matthews correlation coefficient, are meticulously examined to gauge its efficacy.

Despite the challenges encountered, such as the inherent limitations of clustering algorithms and the need for nuanced adjustments in real-world scenarios, the machine learning phase emerges as a promising tool in distinguishing authentic location submissions from deceptive

ones. However, the protocol doesn't merely rely on machine learning; it incorporates a third phase, the location-proof mechanism, to further fortify the verification process.

Through a comprehensive exploration of location spoofing methods and the vulnerabilities of GPS reporting, GSM reporting, LAN reporting, WAN reporting, and Bluetooth technologies, the thesis underscores the need for multifaceted solutions. It critically analyzes the strengths and weaknesses of various technologies, considering aspects like encryption, jamming resistance, and real-time tracking capabilities.

The thesis also acknowledges the limitations of the proposed protocol, particularly in scenarios where users intentionally manipulate their location data. It addresses the trade-offs between security and privacy, emphasizing the need to strike a delicate balance. The exploration of end-to-end encryption and its vulnerabilities, such as man-in-the-middle attacks, adds a layer of complexity to the discussion, highlighting the perpetual challenges in securing data transmission.

Furthermore, the thesis introduces a machine learning phase that utilizes unsupervised learning techniques, specifically the k-means clustering algorithm, to assess the authenticity of user-submitted locations. The methodology involves clustering visited places based on their geographical proximity, creating unique identifiers for each cluster, and subsequently classifying new locations as either real or fake based on their alignment with these clusters. The results, while promising, reveal certain challenges associated with the algorithm's performance, especially in scenarios with imbalanced datasets.

The thesis concludes by emphasizing the need for continuous refinement and adaptation of the proposed protocol. It acknowledges the complexities inherent in securing location-based systems and highlights the importance of addressing emerging threats and technological advancements. The insights gained from this exploration contribute to the ongoing discourse on

enhancing the security of location-based applications, paving the way for future research endeavors in this evolving field.

5.2 Future Recommendations

Enhancement of Machine Learning Model:

Further refinement and enhancement of the machine learning model, particularly the k-means clustering algorithm, are recommended. This could involve exploring alternative unsupervised learning techniques or hybrid models to address the challenges associated with imbalanced datasets and the inherent limitations of clustering algorithms.

Integration of Advanced Encryption Techniques:

Investigate and integrate advanced encryption techniques to bolster the security of location-based systems. This includes exploring post-quantum cryptography methods and evaluating their applicability to safeguard sensitive location data against emerging threats.

Real-World Testing and Validation:

Conduct extensive real-world testing to validate the proposed protocol's effectiveness in diverse scenarios. Collaborate with users from different geographical locations, demographics, and usage patterns to ensure the robustness and adaptability of the system.

User Education and Awareness:

Develop educational initiatives and awareness campaigns to inform users about the importance of location privacy and the potential risks associated with location-based applications. Promote responsible use of such applications and encourage users to adopt recommended security measures.

Collaboration with Industry Stakeholders:

Collaborate with industry stakeholders, including mobile device manufacturers, application developers, and cybersecurity experts, to implement standardized security measures at the hardware and software levels. This collaborative effort can contribute to creating a more secure environment for location-based services.

Continuous Monitoring and Threat Intelligence:

Establish mechanisms for continuous monitoring of evolving threats in the cybersecurity landscape. Implement threat intelligence systems to stay abreast of new techniques employed by malicious actors to manipulate location data and adapt the protocol accordingly.

Research on Location Verification Techniques:

Dedicate research efforts to address the challenge of location verification, especially in preventing users from providing fake location data. Investigate novel techniques, such as zero-knowledge proofs or distributed ledger technologies, to verify user locations without compromising privacy.

Usability Studies and User Feedback:

Conduct usability studies and seek feedback from end-users to understand their experience with the proposed protocol. Identify any usability challenges, concerns, or suggestions for improvement and iteratively enhance the protocol based on user input.

Regulatory Compliance and Ethical Considerations:

Stay informed about evolving privacy regulations and standards related to location-based services. Ensure that the protocol aligns with these regulations and incorporates ethical considerations, respecting user privacy while providing a secure user experience.

Exploration of Emerging Technologies:

Explore the potential integration of emerging technologies, such as blockchain, for enhancing the security and transparency of location-based systems. Evaluate the feasibility and advantages of incorporating these technologies into the existing protocol.

By addressing these recommendations, future research and development efforts can contribute to the ongoing evolution of secure and privacy-preserving location-based systems, ensuring that they remain resilient against emerging threats and provide users with a trustworthy and seamless experience.

References

- [1]Q. Huang, J. Du, and G. Yan, “Privacy-Preserving Spatio-Temporal Keyword Search for Outsourced Location-Based Services | IEEE Journals & Magazine | IEEE Xplore,” *ieeexplore.ieee.org*, Jun. 11, 2021. <https://ieeexplore.ieee.org/abstract/document/9453150> (accessed Nov. 20, 2023).
- [2]K. Edemacu, H. K. Park, and B. Jang, “Privacy Provision in Collaborative Ehealth With Attribute-Based Encryption: Survey, Challenges and Future Directions | IEEE Journals & Magazine | IEEE Xplore,” *ieeexplore.ieee.org*, Jun. 27, 2019. <https://ieeexplore.ieee.org/abstract/document/8747355> (accessed Nov. 20, 2023).
- [3]H. Jiang, J. Li, P. Zhao, F. Zeng, Z. Xiao, and A. Iyengar, “Location Privacy-preserving Mechanisms in Location-based Services,” *ACM Computing Surveys*, vol. 54, no. 1, pp. 1–36, Feb. 2021, doi: <https://doi.org/10.1145/3423165>.
- [4]P. Kanchanadevi, L. Raja, D. Selvapandian, and R. Dhanapal, “An Attribute Based Encryption Scheme with Dynamic Attributes Supporting in the Hybrid Cloud | IEEE Conference Publication | IEEE Xplore,” *ieeexplore.ieee.org*, Oct. 09, 2020. <https://ieeexplore.ieee.org/abstract/document/9243370> (accessed Nov. 20, 2023).
- [5]J. Lin, J. Niu, H. Li, and M. Atiquzzaman, “A Secure and Efficient Location-based Service Scheme for Smart Transportation,” *Future Generation Computer Systems*, vol. 92, pp. 694–704, Mar. 2019, doi: <https://doi.org/10.1016/j.future.2017.11.030>.
- [6]F. Huang, M. Waqas, S. Tu, G. Abbas, and Z. H. Abbas, “A revocable and outsourced multi-authority attribute-based encryption scheme in fog computing,” *Computer Networks*, p. 108196, May 2021, doi: <https://doi.org/10.1016/j.comnet.2021.108196>.
- [7]A. Shankar, P. Pandiaraja, K. Sumathi, T. Stephan, and P. Sharma, “Privacy preserving E-voting cloud system based on ID based encryption,” *Peer-to-Peer Networking and Applications*, Aug. 2020, doi: <https://doi.org/10.1007/s12083-020-00977-4>.
- [8]L. Xue, Y. Yu, Y. Li, M. H. Au, X. Du, and B. Yang, “Efficient attribute-based encryption with attribute revocation for assured data deletion,” *Information Sciences*, vol. 479, pp. 640–650, Apr. 2019, doi: <https://doi.org/10.1016/j.ins.2018.02.015>.
- [9]T. Khoachev, “A Brief Review on Attribute-Based Encryption Approaches,” *Social Science Research Network*, Jul. 31, 2023. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4526403 (accessed Nov. 20, 2023).
- [10]J. Shi, Q. Yu, Y. Yu, L. Wang, and W. Zhang, “Privacy protection in social applications: A ciphertext policy attribute-based encryption with keyword search,” *International*

Journal of Intelligent Systems, vol. 37, no. 12, pp. 12152–12168, Sep. 2022, doi: <https://doi.org/10.1002/int.23080>.

[11]L. WU, H. Wang, Z. Liu, and W. Meng, “Accurate Range Query With Privacy Preservation for Outsourced Location-Based Service in IoT | IEEE Journals & Magazine | IEEE Xplore,” *ieeexplore.ieee.org*, Apr. 24, 2021. <https://ieeexplore.ieee.org/abstract/document/9385397> (accessed Nov. 20, 2023).

[12]J. Zhou, Z. Cao, Z. Qin, and X. Dong, “LPPA: Lightweight Privacy-Preserving Authentication From Efficient Multi-Key Secure Outsourced Computation for Location-Based Services in VANETs | IEEE Journals & Magazine | IEEE Xplore,” *ieeexplore.ieee.org*, Jun. 14, 2019. <https://ieeexplore.ieee.org/abstract/document/8736796> (accessed Nov. 20, 2023).

[13]X. Li, Y. Zhu, and J. Wang, “Highly Efficient Privacy Preserving Location-Based Services with Enhanced One-Round Blind Filter | IEEE Journals & Magazine | IEEE Xplore,” *ieeexplore.ieee.org*, May 06, 2019. <https://ieeexplore.ieee.org/abstract/document/8756044> (accessed Nov. 20, 2023).

[14]S. Xu *et al.*, “Efficient ciphertext-policy attribute-based encryption with blackbox traceability,” vol. 538, pp. 19–38, Oct. 2020, doi: <https://doi.org/10.1016/j.ins.2020.05.115>.

[15]I. Denisow, S. Zickau, F. Beierle, and A. Küpper, “Dynamic Location Information in Attribute-Based Encryption Schemes | IEEE Conference Publication | IEEE Xplore,” *ieeexplore.ieee.org*, Jan. 07, 2016. <https://ieeexplore.ieee.org/abstract/document/7373250> (accessed Nov. 20, 2023).

[16]M. Portnoi and Chien-Chung Shen, “Loc-Auth: Location-enabled authentication through attribute-based encryption | IEEE Conference Publication | IEEE Xplore,” *ieeexplore.ieee.org*, Mar. 05, 2015. <https://ieeexplore.ieee.org/abstract/document/7069321> (accessed Nov. 20, 2023).

[17]M. S. Abolghasemi, M. M. Sefidab, and R. E. Atani, “Using location based encryption to improve the security of data access in cloud computing,” *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Aug. 2013, doi: <https://doi.org/10.1109/icacsi.2013.6637181>.

[18]R. Karimi and M. Kalantari, “Enhancing security and confidentiality on mobile devices by location-based data encryption | IEEE Conference Publication | IEEE Xplore,” *ieeexplore.ieee.org*, Mar. 15, 2012. <https://ieeexplore.ieee.org/abstract/document/6168482> (accessed Nov. 20, 2023).

[19]Y. Borse, D. Patole, and P. Ahirao, “Geo-Encryption: A location based encryption technique for data security | IEEE Conference Publication | IEEE Xplore,” *ieeexplore.ieee.org*,

Jun. 30, 2020. <https://ieeexplore.ieee.org/abstract/document/9129586> (accessed Nov. 20, 2023).

[20]Shih-Hau Fang, Wei-Jia Lai, and Yi-Chung Liang, “An encryption-based approach for protecting privacy in network-based location systems | IEEE Conference Publication | IEEE Xplore,” *ieeexplore.ieee.org*, Sep. 12, 2011.

<https://ieeexplore.ieee.org/abstract/document/6016714> (accessed Nov. 20, 2023).

[21]N. Chen, M. Gerla, D. Huang, and X. Hong, “Secure, selective group broadcast in vehicular networks using dynamic attribute based encryption | IEEE Conference Publication | IEEE Xplore,” *ieeexplore.ieee.org*, Jun. 12, 2019.

<https://ieeexplore.ieee.org/abstract/document/5546877> (accessed Nov. 20, 2023).

[22]B. Manoj, B. Harshad, P. Dhiraj, and B. Pratik, “Location Based Encryption-Decryption Approach for Data Security,” *International Journal of Computer Applications Technology and Research*, vol. 3, pp. 610–611, 2014, Accessed: Nov. 20, 2023. [Online]. Available: <https://ijcatr.com/archives/volume3/issue10/ijcatr03101002.pdf>

[23]G. SRIRAM, B. SRIKANTHREDDY, K. V. SESHADRI, and S. N. SURESH, “Location Based Encryption-Decryption System For Android | IEEE Conference Publication | IEEE Xplore,” *ieeexplore.ieee.org*, Jul. 01, 2019.

<https://ieeexplore.ieee.org/abstract/document/8748555> (accessed Nov. 20, 2023).

[24]Hsien-Chou Liao, P.-C. Lee, Y.-H. Chao, and C.-L. Chen, “A Location-Dependent Data Encryption Approach for Enhancing Mobile Information System Security | IEEE Conference Publication | IEEE Xplore,” *ieeexplore.ieee.org*, Jul. 04, 2017.

<https://ieeexplore.ieee.org/abstract/document/4195213> (accessed Nov. 20, 2023).

[25]H. Zhu, R. Lu, C. Huang, L. Chen, and H. Li, “An Efficient Privacy-Preserving Location-Based Services Query Scheme in Outsourced Cloud | IEEE Journals & Magazine | IEEE Xplore,” *ieeexplore.ieee.org*, Nov. 15, 2015.

<https://ieeexplore.ieee.org/abstract/document/7327242> (accessed Nov. 20, 2023).

[26]P. G. Kolapwar and H. P. Ambulgekar, “Location based data encryption methods and applications | IEEE Conference Publication | IEEE Xplore,” *ieeexplore.ieee.org*, Dec. 03, 2015.

<https://ieeexplore.ieee.org/abstract/document/7342632> (accessed Nov. 20, 2023).

[27]Anju S and J. Joseph, “Location Based Service Applications to secure locations with dual encryption | IEEE Conference Publication | IEEE Xplore,” *ieeexplore.ieee.org*, Aug. 13, 2015. <https://ieeexplore.ieee.org/abstract/document/7193061> (accessed Nov. 20, 2023).

[28]J. Shao, R. Lu, and X. Lin, “FINE: A fine-grained privacy-preserving location-based service framework for mobile devices | IEEE Conference Publication | IEEE Xplore,” *ieeexplore.ieee.org*, Jul. 08, 2014. <https://ieeexplore.ieee.org/abstract/document/6847945>

(accessed Nov. 20, 2023).

[29]Z. Chen, J. Nie, Z. Li, W. Susilo, and C. Ge, “Geometric Searchable Encryption for Privacy-Preserving Location-Based Services | IEEE Journals & Magazine | IEEE Xplore,” *ieeexplore.ieee.org*, Feb. 27, 2023. <https://ieeexplore.ieee.org/abstract/document/10054122> (accessed Nov. 20, 2023).

[30]J. D. Bokefode, A. S. Bhise, P. A. Satarkar, and D. G. Modani, “Developing a Secure Cloud Storage System for Storing IoT Data by Applying Role Based Encryption,” *Procedia Computer Science*, vol. 89, pp. 43–50, 2016, doi: <https://doi.org/10.1016/j.procs.2016.06.007>.

[31]S. G. Weber, “Designing a Hybrid Attribute-Based Encryption Scheme Supporting Dynamic Attributes,” *ePrint IACR*, 2013. <https://eprint.iacr.org/2013/219> (accessed Nov. 20, 2023).

[32]S. G. Weber, “A Hybrid Attribute-Based Encryption Technique Supporting Expressive Policies and Dynamic Attributes,” *Information Security Journal: A Global Perspective*, vol. 21, no. 6, pp. 297–305, Jan. 2012, doi: <https://doi.org/10.1080/19393555.2012.738374>.

[33]A. Salim, S. Tripathi, and R. K. Tiwari, “Applying Geo-Encryption and Attribute Based Encryption to Implement Secure Access Control in the Cloud,” *SSRN Electronic Journal*, 2019, doi: <https://doi.org/10.2139/ssrn.3459330>.

[34]A. Salim, S. Tripathi, and R. K. Tiwari, “Applying Geo-Encryption and Attribute Based Encryption to Implement Secure Access Control in the Cloud,” *SSRN Electronic Journal*, 2019, doi: <https://doi.org/10.2139/ssrn.3459330>.

[35]Nur Nabila Mohamed, Hanunah Othman, Mohd Anuar Mat Isa, Nur Afifah Mohd Noor, and Habibah Hashim, “A secure communication in location based services using AES256 encryption scheme,” *ieeexplore.ieee.org*, Oct. 19, 2017. <https://ieeexplore.ieee.org/abstract/document/8074970>

[36]Liang Zhang, Haibin Kan, and Yihao Wang, “Privacy-Preserving AGV Collision-Resistance at the Edge Using Location-Based Encryption | IEEE Journals & Magazine | IEEE Xplore,” *ieeexplore.ieee.org*, Jan. 09, 2023. <https://ieeexplore.ieee.org/abstract/document/10012042> (accessed Nov. 20, 2023).

[37]Marcos Portnoi and Chien-Chung Shen, “Location-aware sign-on and key exchange using attribute-based encryption and Bluetooth beacons | IEEE Conference Publication | IEEE Xplore,” *ieeexplore.ieee.org*, Dec. 12, 2013. <https://ieeexplore.ieee.org/abstract/document/6682750> (accessed Nov. 20, 2023).

[38]A. Al-Fuqaha, O. Al-Ibrahim, and J. Baird, “A mobility model of GPS-based encryption | IEEE Conference Publication | IEEE Xplore,” *ieeexplore.ieee.org*, Jan. 23, 2016. <https://ieeexplore.ieee.org/abstract/document/1577944> (accessed Nov. 20, 2023).

[39]J. A. Akinyele, M. W. Pagano, M. D. Green, C. U. Lehmann, Z. N. J. Peterson, and A. D. Rubin, “Securing electronic medical records using attribute-based encryption on mobile devices,” *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices - SPSM '11*, 2011, doi: <https://doi.org/10.1145/2046614.2046628>.

[40]Yiliang Han, Shuaishuai Zhu, Y. Li, and X. Lin, “APPLSS: Adaptive privacy preserved location sharing scheme based on attribute-based encryption | IEEE Journals & Magazine | IEEE Xplore,” *ieeexplore.ieee.org*, Mar. 23, 2021. <https://ieeexplore.ieee.org/abstract/document/9384503> (accessed Nov. 20, 2023).