SMART KEY ENTRY SYSTEM

24-25J-140

Final Report

Jananga Chiran Wickramaarchchi – IT21369810

B.Sc. (Hons) in Information Technology specialized in Cyber Security

Department of Computer System Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

May 2025

SMART KEY ENTRY SYSTEM

24-25J-140

Jananga Chiran Wickramaarchchi – IT21369810

B.Sc. (Hons) in Information Technology specialized in Cyber Security

Department of Computer System Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

May 2025

Declaration

I declare that this is my own work, and this dissertation does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant to Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation in whole or part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Group Member Name	Student ID	Signature
Wickramaarachchi J.C.	IT21369810	

……………………………….. ………..…………………….

(Mr. K.Y. Abeywardena) Date

Signature of the supervisor

Acknowledgement

First, I would like to express my deep gratitude to those individuals and institutions that have made this research project possible. Above all, I want to thank my supervisor for being such an enlightening guide through the research and writing phases of the project, which greatly helped in making this project feasible. Your mentorship provided clarity in seeking a path through complex hurdles to achieve your goals related to this project. I am also thankful to the co-supervisor for his very valuable comments and continuous encouragement, which really helped shape the research methodologies and added value to this work.

I would also like to extend my appreciation to the departmental staff for administrative support and for access to necessary resources and tools in the conduct of this research. My thanks go particularly to the technical team that supported me during data collection and in integrating the technology applied in this project. Lastly, I also want to thank my friends and colleagues because through discussions and sharing experiences, it created an enabling learning environment that enriched the development of this research. The support and encouragement have kept me going through this journey. I thank everyone who was involved in the successful execution of this project.

Abstract

The increasing integration of digital technologies in the automotive industry has heightened the vulnerability of traditional Remote Keyless Entry (RKE) systems to attacks such as fixed code exploitation, replay attacks, and brute-force attempts. This research introduces a novel smartphone-based smart key fob application designed to enhance vehicle security and mitigate these threats by leveraging advanced cryptographic methods and proximity-based authentication. The proposed system replaces physical key fobs with a Flutter-based Android application, utilizing Near Field Communication (NFC) for secure door unlocking, ensuring physical proximity, and reducing remote attack risks. Key features include robust user authentication through email, phone verification, and biometric methods, alongside Vehicle Identification Number (VIN) decoding and stolen vehicle checks to prevent unauthorized access. The application implements Role-Based Access Control (RBAC) with time-limited access sharing, enabling vehicle owners to manage permissions flexibly. A risk assessment model, incorporating a machine learning autoencoder, detects anomalies in access patterns based on user role, location, and time, flagging suspicious activities for real-time owner alerts. Secure communication is achieved through AES-256 GCM encryption, challenge-response authentication, and JSON Web Tokens (JWTs), ensuring protection against interception and replay attacks. The system is supported by a Raspberry Pi-based hardware setup, integrating NFC, GPS, and real-time clock modules for offline functionality. This research demonstrates a comprehensive, user-friendly solution that significantly enhances vehicle security, offering a scalable framework for modern automotive access systems.

Table of Contents

List of Figures

List of Tables

List Of Abbreviations

Abbreviations	Description

[1] INTRODUCTION

1.1. Background & Literature Survey

1.1.1 Background

The automotive industry has undergone a transformative shift in recent years, driven by the integration of digital technologies into vehicle systems. This evolution has introduced advanced features such as remote start, GPS tracking, and wireless diagnostics, marking progression toward more connected and intelligent vehicles. However, this increased connectivity has also expanded the attack surface for vehicle access systems, particularly traditional Remote Keyless Entry (RKE) systems that rely on physical key fobs. These key fobs, which primarily use Radio Frequency (RF) communication, offer convenience by allowing users to unlock and start vehicles without a physical key. Despite their widespread adoption, RF-based key fobs are inherently vulnerable to a range of cyber-attacks, including fixed code exploitation, replay attacks, and brute-force attempts, due to their limited computational power and lack of robust encryption mechanisms. According to the Global Automotive Cybersecurity Report 2021, key fobs were identified as the second most common attack vector between 2010 and 2021, with 95% of tested vehicles found susceptible to replay attacks [1][2]. These vulnerabilities have been exploited in real-world scenarios, such as the replay attack vulnerability (CVE-2019-20626) affecting Honda and Acura models in 2020, highlighting the urgent need for more secure alternatives [3].

To address these security gaps, some vehicle manufacturers have developed proprietary smartphone applications that leverage more secure communication protocols, such as Bluetooth Low Energy (BLE) and internet-based commands, to manage vehicle access. However, these solutions often suffer from significant limitations, including their vendor-specific nature, which restricts interoperability across different vehicle brands, and the continued inclusion of physical key fobs, which perpetuates the underlying vulnerabilities of RF-based systems [4]. Furthermore, attacks like RollJam and RollBack, which exploit rolling code systems and code stack resets, respectively, have demonstrated the persistent challenges in securing RKE systems, especially in vehicles where firmware updates are not feasible [5][6]. The growing prevalence of vehicle-related cyber threats underscores the need for a universal, smartphone-based solution that can mitigate these vulnerabilities while offering enhanced security features and user convenience.

Recent studies have explored various approaches to improve vehicle access security. For instance, research on the MSP430 microcontroller has proposed wireless automotive keys with rights-sharing capabilities, focusing on cryptographic enhancements but lacking a mobile application for remote management [7]. Similarly, the PRESTvO framework has advanced smartphone-based vehicle access by incorporating features like engine start and speed limiting, yet it fails to integrate VIN verification for stolen vehicle checks, a critical security feature [8]. Other studies have emphasized two-factor authentication for connected vehicles, improving security but often neglecting comprehensive access control mechanisms like Role-Based Access Control (RBAC) [9]. These gaps in existing research highlight the need for a holistic solution that combines robust encryption, universal compatibility, and intelligent access management to address the evolving threats to vehicle security.

This research proposes a smartphone-based smart key fob application designed to replace traditional RKE systems, utilizing NFC for proximity-based access, advanced cryptographic methods like AES-256 GCM, and a machine learning-driven risk assessment model to detect anomalies. By integrating features such as biometric authentication, VIN verification, and time-sensitive key sharing, the proposed system aims to provide a secure, scalable, and user-friendly alternative for vehicle access management, catering to individual owners, car rental companies, and fleet operators.

1.1.2 Literature Survey

The growing prevalence of cyber threats targeting vehicle access systems has spurred significant research into enhancing the security of Remote Keyless Entry (RKE) systems. Several studies have explored vulnerabilities in traditional key fobs and proposed solutions to mitigate these risks, focusing on cryptographic improvements, smartphone-based access, and advanced authentication mechanisms. However, gaps in these approaches highlight the need for a more comprehensive and universally compatible solution, which this research aims to address.

One notable study by Kamkar (2015) introduced the RollJam attack, which exploits vulnerabilities in rolling code systems used by modern key fobs. Presented at Defcon 23, this attack demonstrated how attackers can capture valid codes by jamming the signal between the key fob and the vehicle, later replaying the captured code to gain unauthorized access. The study emphasized the asynchronous nature of rolling code communication as a critical weakness, underscoring the need for more robust authentication protocols that prevent code capture and replay [10]. While this research provided valuable insights into the limitations of rolling code systems, it did not propose a practical solution for widespread adoption across different vehicle brands.

Another significant contribution came from Cosentino (2022), who analyzed the broader spectrum of RKE vulnerabilities, including fixed code exploitation, replay attacks, and brute-force attempts. The study detailed how fixed code systems, used in early RKE implementations, transmit a static authentication code that can be easily intercepted and replayed by attackers using inexpensive hardware. Cosentino also highlighted the susceptibility of these systems to brute-force attacks due to their limited key space, further compounding their insecurity. The research called for the adoption of dynamic, time-sensitive credentials and proximity-based authentication to mitigate these threats, laying the groundwork for smartphone-based solutions [11].

In response to these vulnerabilities, some researchers have explored smartphone-based access systems as a more secure alternative to traditional key fobs. A study by Groza et al. (2017) proposed a wireless automotive key system with rights-sharing capabilities, implemented on the MSP430 microcontroller. This system focuses on cryptographic enhancements, using pre-shared keys to secure communication and enable access delegation. However, the absence of a mobile application for remote management limited its usability, as users could not easily manage access permissions or monitor vehicle activity in real time [7]. This gap highlights the need for a user-friendly interface that integrates seamlessly with modern smartphones.

Similarly, the PRESTvO framework by G. et al. (2020) advanced smartphone-based vehicle access by enabling features such as engine start, speed limiting, and fuel monitoring through a mobile application. The system utilized Bluetooth Low Energy (BLE) for communication and incorporated privacy-preserving techniques to protect user data. Despite its comprehensive scope, PRESTvO lacked a VIN verification mechanism to check for stolen vehicles, a critical oversight that could allow unauthorized access to stolen vehicles. Additionally, the system's reliance on BLE without additional proximity checks made it potentially vulnerable to relay attacks [8]. This underscores the importance of integrating multiple layers of authentication, such as NFC-based proximity verification, to enhance security.

Further advancements in vehicle security have been made through the integration of multi-factor authentication (MFA). Karacali et al. (2024) proposed a two-factor authentication (2FA) system for connected vehicles, combining a mobile application with a secondary authentication method, such as a PIN or biometric verification. This approach significantly improved security by requiring multiple credentials for access, reducing the risk of unauthorized entry even if one factor was compromised. However, the study did not address the need for fine-grained access control mechanisms, such as Role-Based Access Control (RBAC), which are essential for managing permissions in scenarios involving multiple users, such as car-sharing or fleet management [9]. This limitation points to the need for a system that combines strong authentication with flexible access management.

The use of machine learning (ML) for anomaly detection in vehicle access systems has also gained traction in recent research. A study by Rosenstatter (2022) explored countermeasures to relay attacks, proposing the use of ML models to detect abnormal access patterns based on contextual data, such as location and time of access. The research demonstrated the effectiveness of ML in identifying suspicious activities, such as access attempts in high-theft areas or at unusual times. However, the study focused primarily on relay attacks and did not address other common threats, such as replay or brute-force attacks, nor did it integrate ML with a broader risk assessment framework for real-time decision-making [12]. This suggests an opportunity to develop a more comprehensive risk model that leverages ML to detect a wider range of anomalies while informing access control decisions.

Additionally, the Global Automotive Cybersecurity Report 2021 by Upstream provided a broader perspective on vehicle security trends, revealing that 95% of tested vehicles were vulnerable to replay attacks due to inadequate encryption and authentication mechanisms. The report also noted a significant rise in vehicle-related cyber incidents between 2010 and 2021, with key fobs identified as a primary attack vector. This data emphasizes the urgent need for solutions that move beyond RF-based communication, adopting more secure protocols like BLE and NFC, and incorporating advanced cryptographic methods to protect against interception and replay [2].

Despite these advancements, existing solutions often fall short in addressing the full spectrum of RKE vulnerabilities while ensuring universal compatibility and user convenience. Manufacturer-specific applications, while more secure than traditional key fobs, are limited by their lack of interoperability across different vehicle brands and their reliance on physical key fobs as a fallback, which perpetuates security risks [4]. Moreover, many proposed systems lack the integration of intelligent risk assessment models or proximity-based authentication, leaving them vulnerable to remote attacks. This research aims to bridge these gaps by developing a smartphone-based smart key fob application that combines NFC for proximity-based access, AES-256 GCM encryption for secure communication, RBAC for flexible access management, and a machine learning-driven risk assessment model to detect and respond to suspicious activities in real time.

1.1.3 Research Gap

Despite significant advancements in vehicle security research, several critical gaps remain in addressing the vulnerabilities of Remote Keyless Entry (RKE) systems and providing a comprehensive, universally compatible solution for modern automotive access. Existing studies have made notable progress in identifying and mitigating specific threats, such as replay attacks, RollJam attacks, and brute-force attempts on fixed codes, but they often fall short in delivering a holistic system that integrates robust encryption, flexible access control, intelligent risk assessment, and universal compatibility across vehicle brands. This research aims to address these gaps by developing a smartphone-based smart key fob application that combines advanced security features with user-centric design.

One prominent gap in the literature is the lack of universal compatibility in existing smartphone-based vehicle access solutions. Many manufacturer-specific applications, such as those developed by Tesla, are designed to work exclusively with their own vehicle models, limiting their usability for users with vehicles from multiple brands [4]. This proprietary approach alienates a significant portion of the market, particularly fleet operators and car rental companies that manage diverse vehicle portfolios. Moreover, these applications often retain physical key fobs as a fallback, perpetuating the vulnerabilities associated with RF-based communication, such as susceptibility to replay and RollJam attacks [10][11]. A universal solution that eliminates the need for physical key fobs while ensuring compatibility across different vehicle brands is essential to address this limitation.

Another critical gap is the absence of comprehensive access control mechanisms in many proposed systems. While studies like Karacali et al. (2024) have successfully implemented two-factor authentication (2FA) to enhance security for connected vehicles, they often lack fine-grained access control frameworks like Role-Based Access Control (RBAC) [9]. RBAC is crucial for managing permissions in scenarios involving multiple users, such as car-sharing or temporary access for guests, where different roles (e.g., owner, friend, family) require distinct levels of access. The PRESTvO framework by G. et al. (2020) similarly falls short in this regard, focusing on features like engine start and speed limiting but neglecting the need for role-based permissions and time-sensitive key sharing, which are vital for secure and flexible access management [8]. This research addresses this gap by integrating RBAC and time-limited access sharing, enabling vehicle owners to define and manage permissions effectively.

The integration of intelligent risk assessment and anomaly detection is another area where existing research is limited. While Rosenstatter (2022) explored the use of machine learning (ML) to detect relay attacks by analyzing contextual data like location and time, the study did not extend its ML model to address other common threats, such as replay or brute-force attacks, nor did it incorporate a broader risk assessment framework for real-time decision-making [12]. Additionally, many systems lack the ability to flag suspicious activities based on user roles or high-theft locations, which are critical for identifying unauthorized access attempts. This research bridges this gap by implementing a machine learning-driven autoencoder to detect anomalies in access patterns, combined with a risk calculation model that evaluates factors such as user role, location, and time, providing real-time alerts to vehicle owners.

Furthermore, existing solutions often fail to incorporate robust anti-theft mechanisms, such as VIN verification for stolen vehicle checks. The PRESTvO framework, for instance, enables smartphone-based access but does not include a mechanism to verify the vehicle’s legal status before granting access, potentially allowing stolen vehicles to be used without detection [8]. This is a significant oversight, as integrating VIN verification with a stolen vehicle database can add an essential layer of security, particularly for fleet operators and individual owners concerned about vehicle theft. This research addresses this gap by implementing VIN decoding and stolen vehicle checks during the vehicle registration process, ensuring that only legitimate vehicles can be accessed through the application.

Finally, the reliance on RF-based communication in many existing systems, even those with digital enhancements, leaves them vulnerable to remote attacks. Studies like Cosentino (2022) have highlighted the ease with which fixed code systems can be exploited through replay and brute-force attacks, emphasizing the need for proximity-based authentication and dynamic credentials [11]. While some systems have adopted Bluetooth Low Energy (BLE) for communication, they often lack additional proximity checks, making them susceptible to relay attacks [8]. The proposed system mitigates this by using Near Field Communication (NFC) for door unlocking, requiring physical proximity, and employing AES-256 GCM encryption with time-sensitive credentials to prevent interception and replay. Additionally, the system’s multi-layered authentication process, including challenge-response mechanisms and encrypted JSON Web Tokens (JWTs), ensures robust protection against remote exploitation.

The following table summarizes the research gaps identified in existing studies and how this work addresses them:

Feature	**Groza et al. (2017) [7]**	**PRESTvO (2020) [8]**	**Karacali et al. (2024) [9]**	**This Work**
Universal Compatibility	✗	✗	✗	✓
Role-Based Access Control	✗	✗	✗	✓
Time-Sensitive Key Sharing	✗	✗	✗	✓
VIN Verification	✗	✗	✗	✓
ML-Driven Anomaly Detection	✗	✗	✗	✓
Proximity-Based Access (NFC)	✗	✗	✗	✓

Table 1 Research Gap Summary

This research fills these gaps by developing a smartphone-based smart key fob application that offers universal compatibility, integrates RBAC with time-sensitive key sharing, employs ML-driven anomaly detection, and ensures robust security through NFC-based access and advanced cryptographic methods. By addressing these deficiencies, the proposed system provides a comprehensive solution for secure and flexible vehicle access management.

1.1.4 Research Problem

The rapid integration of digital technologies into the automotive industry has significantly enhanced vehicle functionality, but it has also exposed vehicle access systems to a growing array of cyber threats. Traditional Remote Keyless Entry (RKE) systems, which rely on physical key fobs and Radio Frequency (RF) communication, are particularly vulnerable to attacks such as fixed code exploitation, replay attacks, and brute-force attempts, as well as more sophisticated threats like RollJam and RollBack attacks. These vulnerabilities, combined with the limitations of existing digital solutions, present significant security challenges that undermine the safety and reliability of vehicle access systems. This research identifies two primary problems that highlight the urgent need for a more secure, universally compatible, and intelligent vehicle access solution.

1. **Vulnerabilities in Traditional Physical Key Fobs**

   Traditional physical key fobs, which operate using RF communication, are highly susceptible to a range of cyber-attacks due to their limited computational power and lack of advanced cryptographic measures. Fixed code systems, used in early RKE implementations, transmit static authentication codes that can be easily intercepted and replayed by attackers using inexpensive hardware, as demonstrated by Cosentino (2022) [11]. The Global Automotive Cybersecurity Report 2021 further revealed that 95% of tested vehicles were vulnerable to replay attacks, with key fobs identified as the second most common attack vector between 2010 and 2021 [2]. Additionally, rolling code systems, designed to mitigate replay attacks, remain susceptible to RollJam attacks, which exploit the asynchronous nature of communication to capture and replay valid codes [10]. RollBack attacks, unveiled at BlackHat USA 2022, further exacerbate these vulnerabilities by allowing attackers to reset the vehicle’s code stack, enabling repeated unauthorized access [13]. These attacks are particularly difficult to mitigate in vehicles where firmware updates are not feasible, leaving a significant portion of the automotive market exposed to theft and unauthorized access [12]. The reliance on RF communication, which lacks proximity-based authentication, also makes these systems vulnerable to remote exploitation, posing a substantial risk to vehicle security.

2. **Limitations of Manufacturer-Specific Digital Key Fob Applications**

   In response to the vulnerabilities of physical key fobs, some manufacturers have developed smartphone-based digital key fob applications that leverage more secure communication protocols, such as Bluetooth Low Energy (BLE) and internet-based commands. However, these solutions are often vendor-specific, limiting their interoperability across different vehicle brands and models. For instance, Tesla's digital key system is designed exclusively for Tesla vehicles, restricting its usability for users with vehicles from multiple manufacturers [4]. This lack of universal compatibility is a significant barrier for fleet operators, car rental companies, and individual owners with diverse vehicle portfolios. Moreover, these applications typically include a physical key fob as a fallback mechanism, perpetuating the inherent security flaws of RF-based systems, such as susceptibility to replay and RollJam attacks [10][11]. The absence of proximity-based authentication in many BLE-based systems also leaves them vulnerable to relay attacks, where attackers can extend the communication range to gain unauthorized access [8]. Additionally, these applications often lack advanced access control mechanisms, such as Role-Based Access Control (RBAC) and time-sensitive key sharing, which are essential for managing permissions in multi-user scenarios like car-sharing or temporary access [9]. The failure to integrate intelligent risk assessment models further limits their ability to detect and respond to suspicious activities, such as access attempts in high-theft areas or at unusual times [12].

These problems highlight the need for a smartphone-based vehicle access system that eliminates the vulnerabilities of physical key fobs, ensures universal compatibility, and incorporates advanced security features. The proposed smart key fob application addresses these challenges by utilizing Near Field Communication (NFC) for proximity-based access, implementing AES-256 GCM encryption with time-sensitive credentials, and integrating RBAC with a machine learning-driven risk assessment model to detect anomalies and enhance security.

1.1.5 Research Objectives

1.1.5.1 Main Objectives

The primary objective of this research is to develop a smartphone-based smart key fob application that replaces traditional physical key fobs, addressing the vulnerabilities of Remote Keyless Entry (RKE) systems and providing a secure, universally compatible solution for vehicle access. By leveraging the advanced computational capabilities of smartphones, the application aims to enhance security through robust encryption, proximity-based authentication, and intelligent risk assessment, while offering flexible access management for diverse user scenarios. Specifically, the main objectives are:

- **Mitigate Vulnerabilities of Traditional RKE Systems:** The application seeks to eliminate the security risks associated with physical key fobs, such as fixed code exploitation, replay attacks, and brute-force attempts, by utilizing Near Field Communication (NFC) for proximity-based access and AES-256 GCM encryption for secure communication. This ensures that access is granted only to authorized users in close physical proximity to the vehicle, significantly reducing the risk of remote attacks [11][13].

- **Ensure Universal Compatibility Across Vehicle Brands:** Unlike manufacturer-specific digital key solutions, this application is designed to be compatible with a wide range of vehicle brands and models, catering to individual owners, car rental companies, and fleet operators. By integrating Vehicle Identification Number (VIN) decoding and verification, the system ensures seamless registration and access management for diverse vehicle portfolios, overcoming the interoperability limitations of existing solutions [4][8].

- **Enhance Security with Advanced Authentication and Risk Assessment:** The system aims to implement a multi-layered security framework that includes biometric authentication, challenge-response mechanisms, and time-sensitive credentials to prevent unauthorized access. Additionally, a machine learning-driven risk assessment model, using an autoencoder, will detect anomalies in access patterns based on user role, location, and time, providing real-time alerts to vehicle owners for suspicious activities [9][12].

- **Provide Flexible Access Management for Multi-User Scenarios:** The application will incorporate Role-Based Access Control (RBAC) and time-sensitive key sharing, enabling vehicle owners to define and manage access permissions for different user roles (e.g., owner, friend, family) with specified time windows. This feature addresses the needs of car-sharing, temporary access, and fleet management, offering a user-friendly and secure solution for access delegation [7][9].

By achieving these objectives, the proposed smart key fob application aims to set a new standard for vehicle access security, combining advanced cryptographic methods, intelligent anomaly detection, and universal compatibility to meet the evolving needs of the automotive industry.

1.1.5.2 Specific Objectives

To achieve the main objectives of developing a secure, universally compatible smartphone-based smart key fob application, this research outlines several specific objectives that focus on the design, implementation, and evaluation of key system components. These sub-objectives ensure the system addresses the identified vulnerabilities, enhances security, and provides a user-friendly experience for diverse use cases.

1.1.5.2.1 Design and Develop a Flutter-Based Android Application for Vehicle Access.

The objective is to architect and build a Flutter-based Android application that serves as a digital key fob, replacing traditional physical key fobs with a more secure and feature-rich solution. The application will provide an intuitive user interface for managing vehicle access, incorporating features like user authentication, vehicle registration, and access sharing, while ensuring compatibility across a wide range of Android devices [14].

1.1.5.2.2 Proximity-based Door Unlocking

This objective focuses on utilizing Near Field Communication (NFC) for proximity-based door unlocking, ensuring that access is granted only when the user is physically close to the vehicle. Additionally, the system will implement AES-256 GCM encryption with time-sensitive credentials to secure communication between the smartphone, server, and vehicle hardware unit, protecting against interception, replay attacks, and brute-force attempts [11][13].

1.1.5.2.3 Integrate Robust User Authentication and Role-Based Access Control (RBAC):

The goal is to develop a comprehensive user authentication system that includes email and phone verification, followed by biometric authentication (e.g., fingerprint or facial recognition) for adding vehicles. The system will also implement RBAC, allowing vehicle owners to define and manage access permissions based on user roles (e.g., owner, friend, family) and share access with time-limited windows, addressing the needs of multi-user scenarios like car-sharing and fleet management [7][9].

1.1.5.2.4 Incorporate VIN Verification and Stolen Vehicle Checks:

This objective aims to integrate a VIN verification system that decodes the Vehicle Identification Number using an external API (e.g., vindecoder.eu) and checks the vehicle against a database of stolen vehicles before granting access. If a vehicle is flagged as stolen, the system will notify authorities, ensuring that only legitimate vehicles can be accessed and enhancing user trust and security [8].

1.1.5.2.5 Develop a Machine Learning-Driven Risk Assessment Model for Anomaly Detection:

The system will implement a risk assessment model that uses a machine learning autoencoder to detect anomalies in vehicle access patterns based on contextual data, such as user role, location, and time of access. The model will calculate a risk score, flagging suspicious activities (e.g., access in high-theft areas or at unusual times) as low, medium, or high risk, and alert the vehicle owner in real time to enhance security [12].

1.1.5.2.6 Simulate Vehicle Hardware Interaction Using Raspberry Pi and NFC Modules:

This objective involves setting up a hardware simulation using a Raspberry Pi 4 as a vehicle control unit, interfaced with a PN532 NFC reader for door unlocking, a NEO-6M GPS module for location tracking, and a DS3231 Real-Time Clock (RTC) for timekeeping. The hardware setup will support offline functionality and validate the system’s feasibility in real-world automotive scenarios, ensuring reliable operation even in limited connectivity environments [15].

[2] METHODOLOGY

2.1 Methodology Description

The Smart Key Entry System is designed to address the vulnerabilities inherent in traditional Remote Keyless Entry (RKE) systems, such as fixed code exploitation, replay attacks, and brute-force attempts, by leveraging a smartphone-based application to provide secure and convenient vehicle access. The methodology adopted for this research involves a systematic approach to designing, developing, and integrating multiple components, including a Flutter-based Android application, a hardware control unit, advanced cryptographic mechanisms, and a machine learning-driven risk assessment model. This subsection provides a detailed description of the system architecture, user workflow, security protocols, hardware setup, and anomaly detection mechanism, ensuring alignment with the research objectives of enhancing vehicle security while maintaining user convenience.

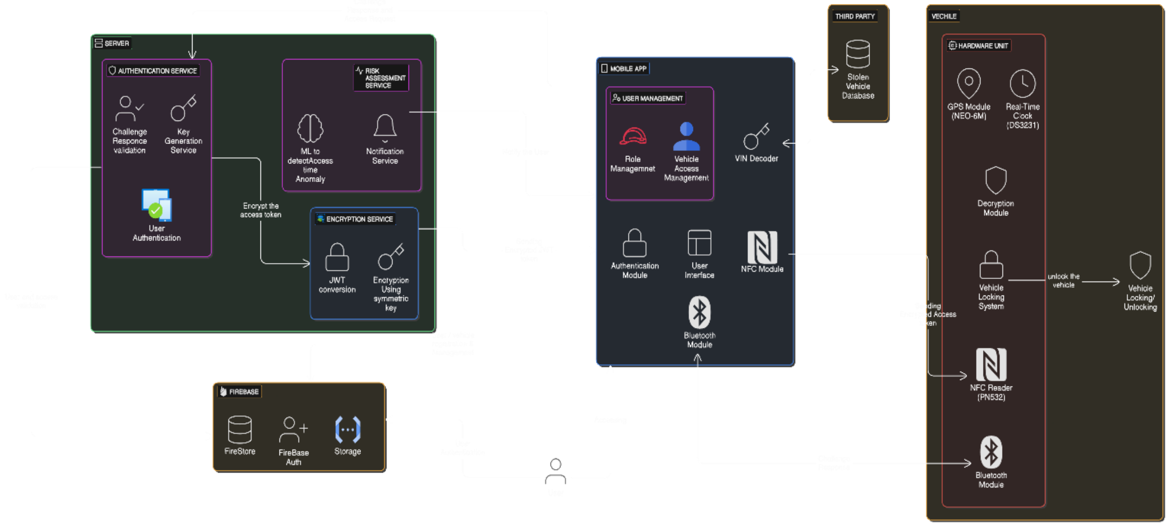

Figure 1 High-Level system diagram

The system architecture is structured around three primary components: the mobile application, the server, and the vehicle control unit. The mobile application, developed using Flutter, serves as the user interface, enabling vehicle owners and authorized users to interact with the system. Flutter was selected for its cross-platform capabilities, its robust support for integrating Near Field Communication (NFC) and Bluetooth functionalities. The server, hosted on a cloud platform (Digital Ocean), handles vehicle registration, encryption and key generation, ensuring secure communication between the app and the vehicle. The vehicle control unit, implemented using a Raspberry Pi 4, interfaces with the vehicles locking system and includes an NFC reader (PN532), a GPS module (NEO-6M), and a real-time clock (DS3231) to support offline functionality. This architecture ensures a seamless flow of secure data, with each component playing a critical role in mitigating RKE vulnerabilities.

The user workflow begins with account creation and vehicle registration, designed to prioritize security and traceability. Users register through the mobile application by providing an email address and phone number, which are verified using one-time passwords (OTPs) sent via email and SMS. This dual-verification process ensures that only legitimate users can access the system. Upon successful verification, users authenticate using biometric methods, such as fingerprint scanning, which adds an additional layer of security by preventing unauthorized access to the app. To register a vehicle, users need to input the Vehicle Identification Number (VIN), which is decoded using an external API to verify the vehicles make, model, and ownership details. The system cross-references the VIN against a stolen vehicle database, flagging any matches and notifying authorities if necessary. This step aligns with the research objective of enhancing vehicle security by preventing the use of stolen vehicles within the system. Once the vehicle is registered, the owner can assign roles to other users, such as "Friend" or "Family," with time-limited access privileges, while retaining full control over user management and access history. Security is a cornerstone of the Smart Key Entry System, with multiple layers of cryptographic mechanisms implemented to protect against common RKE attacks. The system employs AES-256 GCM (Advanced Encryption Standard with Galois/Counter Mode) for data encryption, ensuring confidentiality and integrity during communication between the app, server, and vehicle control unit. AES-256 GCM was chosen for its proven resistance to cryptographic attacks and its ability to provide authenticated encryption, which prevents tampering with transmitted data [16]. A challenge-response authentication mechanism is integrated to mitigate replay attacks. When a user attempts to unlock the vehicle, the app sends a request to the server, which generates a unique challenge. The app must respond with a valid response, computed using a pre-shared key, before the server issues a JSON Web Token (JWT) with time-sensitive credentials. The JWT, valid for a short duration (e.g., 30 seconds), is encrypted and transmitted to the vehicle control unit via Bluetooth, ensuring that intercepted tokens cannot be reused after expiration. Additionally, a double encryption process is employed, where the JWT is encrypted with a pre-shared key and a nonce (a random number used only once), further enhancing security by ensuring that each communication session is unique.

The vehicle control unit, implemented on a Raspberry Pi 4, serves as the interface between the mobile application and the vehicles locking system. Raspberry Pi 4 was selected for its computational capabilities, GPIO (General Purpose Input/Output) pins for hardware interfacing, and support for Python-based programming, which facilitates rapid prototyping and integration [17]. The control unit is equipped with an NFC reader (PN532), which operates at 13.56 MHz and supports ISO/IEC 14443 Type A and B cards, enabling secure proximity-based communication with the user's smartphone. When the user taps their phone on the NFC reader, the encrypted JWT is transmitted, decrypted by the control unit, and validated against the pre-shared key and nonce. If the validation is successful, the control unit sends a signal to the vehicles locking mechanism to unlock the door. The integration of NFC ensures that the user must be near the vehicle (typically within 10 cm), significantly reducing the risk of remote attacks compared to Bluetooth-only systems. To support offline functionality, a real-time clock (DS3231) is incorporated into the control unit, allowing the system to validate time-sensitive keys even when the vehicle is not connected to the server. The DS3231 maintains accurate timekeeping with a battery backup, ensuring that keys expire as intended (e.g., after 48 hours without server synchronization). Additionally, a GPS module (NEO-6M) provides location data, which is used to determine if the vehicle is in a high-theft area, a factor in the risk assessment model. More information is mentioned in the Hardware Setup section.

A critical component of the methodology is the machine learning-driven risk assessment model, which enhances the system's ability to detect and respond to suspicious access patterns. An autoencoder, a type of neural network, is employed for anomaly detection due to its effectiveness in identifying patterns that deviate from normal behavior [18]. The autoencoder is trained on historical access data, including user roles (e.g., Owner, Friend, Family), access times, and locations. The model extracts feature such as the frequency of access, the time of day, and the geographical area, which are used to compute a risk score. The risk assessment model evaluates four primary factors: the user's role, the locations high-theft status, time anomalies, and key validity. For instance, if a "Friend" user attempts to access the vehicle at 2:00 AM in a high theft area, the model flags this as a potential anomaly and assigns a risk level (e.g., High Risk). The risk levels Risk, Low Risk, Medium Risk, and High Risk are determined based on weighted scores, where time anomalies contribute 40% to the overall risk, high-theft areas 30%, user role 20%, and key validity 10%. These weights were determined through iterative testing to prioritize factors most correlated with theft attempts, such as unusual access times. If the risk level exceeds a predefined threshold (e.g., Medium Risk), the system alerts the vehicle owner via the mobile application, allowing them to take preventive action, such as revoking access or notifying authorities. Bluetooth is used for initial communication between the app and the vehicle control unit, allowing the user to initiate the unlock process from a short distance (e.g., 10 meters). This step involves transmitting the encrypted JWT and performing the challenge-response authentication. Once the preliminary authentication is complete, the user must tap their smartphone on the NFC reader to finalize the unlock process, ensuring that physical proximity is required to access the vehicle. This dual approach addresses the trade-off between convenience and security: Bluetooth provides the range needed for user convenience, while NFC ensures that the final unlock step is secure against remote attacks [19]. The methodology was iteratively refined through prototyping and feedback, ensuring that each component software, hardware, and machine learning works cohesively to achieve the research objectives of secure vehicle access and anomaly detection.

2.1.1 Hardware setup

The hardware setup of the Smart Key Entry System forms the backbone of the vehicle control unit, enabling secure interaction between the smartphone application and the vehicles locking mechanism. This setup is designed to address the limitations of traditional Remote Keyless Entry (RKE) systems by integrating advanced hardware components that ensure proximity-based access, offline functionality, and location aware risk assessment. The vehicle control unit is implemented using a Raspberry Pi 4 Model B, which serves as the central processing unit, interfacing with an NFC reader (PN532), a GPS module (NEO-6M), a real-time clock (DS3231), and the vehicles locking mechanism. This subsection provides a comprehensive description of each hardware component, its configuration, and its role in achieving the research objectives of enhancing vehicle security and usability. The Raspberry Pi 4 Model B was selected as the core of the vehicle control unit due to its robust computational capabilities, affordability, and extensive support for hardware interfacing.

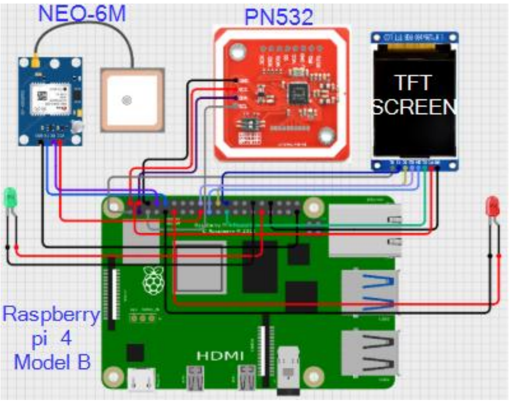

Figure 2 Hardware Unit Wiring Diagram

The Raspberry Pi 4 features a 1.5 GHz quad-core ARM Cortex-A72 processor, 4 GB of RAM, and multiple General-Purpose Input/Output (GPIO) pins, making it ideal for real-time processing and hardware integration [20]. The device runs on Raspberry Pi OS (a Debian-based operating system), which provides a stable environment for developing and deploying the control software. The Raspberry Pi 4 is powered via a 5V USB-C connector, with a power consumption of approximately 3W under normal operation, ensuring energy efficiency within the vehicles electrical system. The control software, written in Python, leverages the Raspberry Pi's GPIO pins to interface with external hardware components, manage communication protocols, and process authentication requests from the mobile application.

A key component of the hardware setup is the PN532 NFC reader, which enables proximity-based communication between the smartphone and the vehicle control unit. The PN532 operates at 13.56 MHz and supports ISO/IEC 14443 Type A and B protocols, making it compatible with most NFC-enabled smartphones [21]. PN532 is connected to the Raspberry Pi 4 via the I2C (Inter-Integrated Circuit) interface, using GPIO pins 3 (SDA) and 5 (SCL) for data transfer, and is powered by the Raspberry Pi's 3.3V pin. The reader is configured to operate in target mode, waiting for an NFC initiator (the smartphone) to establish a connection. When the user taps their NFC enabled smartphone on the reader, the encrypted JSON Web Token (JWT) is transmitted over the NFC channel. The PN532 supports a communication range of approximately 10 cm, ensuring that physical proximity is required for unlocking, thus mitigating the risk of remote attacks such as signal amplification or relay attacks. The NFC reader is mounted near the vehicle's door handle, encased in a weatherproof enclosure to protect it from environmental factors like rain or dust, ensuring reliable operation in real-world conditions. To support location-aware risk assessment, the hardware unit incorporates a NEO-6M GPS module, which provides real-time geographical data to determine if the vehicle is in a high-theft area. The NEO-6M, manufactured by U-blox, offers a position accuracy of 2.5 meters and supports up to 50 channels for satellite tracking [22]. The module is connected to the Raspberry Pi 4 via the UART (Universal Asynchronous Receiver/Transmitter) interface, using GPIO pins 14 (TXD) and 15 (RXD) for serial communication, and is powered by the Raspberry Pis 3.3V pin. The NEO-6M communicates using the NMEA (National Marine Electronics Association) protocol, transmitting data sentences (e.g., GGA, RMC) that include latitude, longitude, and timestamp information. A Python script running on the Raspberry Pi parses these sentences using the "*pynmea2*" library to extract the vehicles coordinates. These coordinates are then cross-referenced with a preloaded database of high-theft areas, which is updated periodically via the server when an internet connection is available. The GPS module is equipped with an external antenna, mounted on the vehicles roof to ensure optimal satellite reception, even in urban environments with potential signal obstructions.

Offline functionality is a critical feature of the Smart Key Entry System, ensuring that the vehicle can be unlocked even in areas with limited or no internet connectivity. This capability is enabled by the DS3231 real-time clock (RTC), which provides accurate timekeeping for validating time-sensitive JWTs. The DS3231 is a highly precise RTC with an integrated temperature-compensated crystal oscillator, offering an accuracy of ±2 ppm (parts per million), equivalent to about 63 seconds of drift per year [23]. The module is connected to the Raspberry Pi 4 via the I2C interface, sharing the same bus as the PN532 (GPIO pins 3 and 5), and is powered by the Raspberry Pi's 3.3V pin. The DS3231 includes a backup battery (CR2032), which allows it to maintain timekeeping even when the vehicle's power is off, ensuring uninterrupted operation. The control software uses the "*Adafruit CircuitPython DS3231"* library to read the current time from the module, which is then compared against the expiry timestamp embedded in the JWT. If the server has not been synced for more than 48 hours, the system defaults to a lockdown mode, requiring the user to reconnect to the server for re-authentication, thus balancing security and usability in offline scenarios.

The integration of these hardware components required careful configuration to ensure seamless operation. The Raspberry Pi 4 boots automatically when the vehicle is powered on, initiating the control software that listens for Bluetooth and NFC communications. The hardware unit is housed in a compact enclosure that can mount under the vehicle's dashboard, with the NFC reader and GPS antenna externally positioned for accessibility and signal reception. This setup not only supports the system's functionality but also aligns with the research objective of creating a practical, secure alternative to traditional RKE systems, capable of operating in diverse real-world conditions.

2.1.2 Secure Communication between Components

The secure communication framework of the Smart Key Entry System is designed to address the vulnerabilities of traditional Remote Keyless Entry (RKE) systems, such as replay attacks, man-in-the-middle attacks, and unauthorized access, by establishing robust data exchange protocols between the mobile app, server, and hardware unit. This subsection provides a detailed description of the communication processes, including authentication, encryption, and token-based access, supported by sequence diagrams that illustrate the interactions. The system leverages AES-256 GCM encryption, challenge-response authentication, and JSON Web Tokens (JWTs) to ensure confidentiality, integrity, and authenticity across all communication channels. The communication between the mobile app and the server initiates the authentication process, forming the first layer of security. As depicted in Figure 3, the app sends the users ID along with an PEM-formatted public key to the server, which stores or updates this key in the users Firestore document. The server then generates a 32-byte random nonce, also stored in Firestore, and transmits it back to the app. The app receives this nonce and signs it with its private key, encoding the signature in DER format. The user ID, along with this signature, is sent back to the server, which retrieves the stored nonce and public key to verify the signature. If valid, the server considers the client authenticated and proceeds. This challenge-response mechanism, detailed in [24], prevents replay attacks by ensuring each session uses a unique, time-sensitive nonce. The server then loads a normal symmetric key from the environment and computes a "*window salt*" based on a two-minute window, which changes every two minutes to enhance security. An ephemeral key, derived from the normal key and valid for the current two-minute window, is used to encrypt a JWT containing the client ID, vehicle ID, and an expiration time (e.g., 5 min) using AES-256 GCM with a 128-bit IV. The encrypted JWT, IV, and authentication tag are returned to the app, completing the secure handoff.

The interaction between the server and the hardware unit, facilitated through the mobile app, ensures that the vehicle control unit receives only authenticated commands. Figure 4 illustrates this process, where the app, after receiving the encrypted JWT, establishes a Bluetooth Low Energy (BLE) connection with the hardware unit. The hardware unit generates a challenge, which is sent to the app via BLE. The app computes a proof based on this challenge and sends it back, which the hardware unit verifies. Upon successful verification, the hardware unit computes a response and sends it to the app, which validates it. The app then sends the encrypted data (JWT, IV, tag) to the hardware unit, which derives an ephemeral key from a normal key stored in its environment. The hardware unit decrypts the JWT using AES-256 GCM, rotating the normal key periodically for added security. This dual layer authentication first with the server and then with the hardware unit mitigates risks of unauthorized access, aligning with the research objective of enhancing vehicle security through advanced cryptographic methods. The NFC channel between the mobile app and the hardware unit serves as the final step for proximity-based unlocking, requiring physical contact within 10 cm to complete the process. After the BLE authentication, the app transmits the encrypted JWT to the hardware units NFC reader (PN532) when the user taps their smartphone. The hardware unit decrypts the JWT using the pre-shared key and nonce, validating its expiry against the real-time clock (DS3231). If valid, the hardware unit sends an unlock signal to the vehicles locking mechanism. This NFC-based communication, combined with the earlier BLE and server interactions, ensures that the system is resistant to remote attacks, a significant improvement over traditional RKE systems that rely on fixed codes or broader-range wireless signals [25]. The sequence diagrams (Figures 3 and 4) provide a visual representation of these interactions. Figure 3, titled "Sequence Diagram: App-to-Server Authentication," illustrates the Nonce-Based Challenge-response and JWT generation process, highlighting the use of public-private key pairs and AES-256 GCM encryption.

Figure 4, titled "Sequence Diagram: App-to-Hardware Unit Authentication," depicts the BLE-based challenge-response and NFC-based unlocking, emphasizing the hardware unit’s role in validating the encrypted JWT. These diagrams, generated based on the systems design, are included in Appendix A for reference and are annotated to clarify each step’s security purpose. To further enhance security, the system employs a double encryption process for the JWT. Initially encrypted with a pre-shared key and a nonce during server-to-app communication, the token is re-encrypted with an ephemeral key during app-to-hardware transmission. This layered approach, combined with the two-minute window for salt and key rotation, reduces the window of vulnerability to interception or decryption attacks. The hardware unit’s offline validation capability, supported by the DS3231 real-time clock, ensures that communication remains secure even without server connectivity, validating keys for up to 48 hours before requiring re-authentication.

2.2 Commercialization aspects of the products

The Smart Key Entry System presents a promising opportunity for commercialization, addressing the growing demand for secure and convenient vehicle access solutions in an era of rising vehicle thefts. The proposed business model offers the mobile application free of charge to attract a wide user base, while the hardware unit is sold at a fixed price to cover initial development and production costs. Additional revenue streams are generated through optional features, including support for extra users beyond a default limit of three, extra vehicles beyond a default limit of three, and future enhancements such as custom roles with fine-tuning and geofencing. This subsection details the market potential, pricing strategy, scalability, partnerships, and future development plans, culminating in a comparative table of subscription plans.

The market potential for the Smart Key Entry System is driven by the increasing prevalence of vehicle thefts and the limitations of traditional Remote Keyless Entry (RKE) systems. According to the National Highway Traffic Safety Administration, approximately 1.6 million vehicles were stolen in the United States in 2023, highlighting the need for advanced security solutions [26]. The system targets automotive manufacturers, aftermarket security providers, and individual vehicle owners, particularly those seeking to upgrade older vehicles with modern, smartphone-based access. The free mobile app, developed using Flutter for cross-platform compatibility, lowers the entry barrier, encouraging adoption among tech-savvy consumers and fleet operators. The hardware unit, comprising a Raspberry Pi 4, NFC reader (PN532), GPS module (NEO-6M), and real-time clock (DS3231), is priced at a fixed value to ensure affordability while covering production costs estimated at $150 per unit based on current component prices and assembly labor.

The pricing strategy incorporates a tiered model to maximize revenue and user satisfaction. The base package includes the free app and one hardware unit, supporting up to three users and three vehicles by default. Additional users beyond three incur a monthly fee of $5 per user, reflecting the increased server load and authentication processing. Similarly, support for additional vehicles beyond three costs $10 per vehicle per month, accounting for the need to manage multiple Vehicle Identification Numbers (VINs) and synchronization requirements. Future updates, not yet implemented but planned for development, include custom roles with fine-tuning allowing owners to define specific access permissions (e.g., time restrictions, location limits) for $8 per role per month and geofencing, which restricts unlocking to pre-defined geographic areas for $12 per month. These features leverage the system’s machine learning risk assessment and GPS capabilities, offering premium security options to justify the additional costs.

Scalability is a key consideration for commercial success, enabling the Smart Key Entry System to expand across diverse markets and vehicle types. The Flutter-based app can be extended to iOS with minimal additional development, broadening the user base to include Apple device owners. The hardware unit’s modular design allows integration with various vehicle models, from sedans to commercial trucks, through adaptable relay modules for locking mechanisms. The cloud-hosted server, built on a scalable platform (Digital Ocean), can handle increased user traffic by adding instances, ensuring performance as the customer base grows. Future scalability plans include partnerships with smart home system providers to enable unified access control (e.g., unlocking both vehicle and home doors via NFC), tapping into the growing Internet of Things (IoT) market projected to reach $1.6 trillion by 2025 [27]. This expansion potential positions the system as a versatile solution for both individual and enterprise customers.

Challenges to commercialization include the initial hardware installation cost and the need for widespread NFC-enabled smartphone adoption. To address these, the system offers a self-installation guide with the hardware unit, reducing labor costs, and targets. Future development will focus on integrating custom roles and geofencing, requiring additional investment in machine learning algorithms and GPS infrastructure.

To provide a clear overview of the commercialization options, Table 2 compares three subscription plans: Basic, Premium, and Enterprise. The Basic plan includes the free app, one hardware unit at $150, and default limits of three users and three vehicles. The Premium plan adds support for six users and six vehicles, with a monthly fee of $10, catering to families or small businesses. The Enterprise plan offers unlimited users and vehicles, custom roles, and geofencing for $30 per month, targeting fleet operators or luxury vehicle owners. These values are estimated based on market research and production costs, ensuring competitiveness and profitability.

Feature	Basic Plan	Premium Plan	Enterprise Plan
App Access	Free	Free	Free
Hardware Unit Cost	$150 (one-time)	$150 (one-time)	$150 (one-time)
Default Users	3	6	Unlimited
Default Vehicles	3	6	Unlimited
Additional User (per month)	$5	$5	Included
Additional Vehicle (per month)	$10	$10	Included
Custom Roles (per month)	Not Available	Not Available	$8 per role
Geofencing (per month)	Not Available	Not Available	$8 per vehicle
Monthly Subscription	$0	$10	$30

Table 2 Comparison of Subscription Plans for Smart Key Entry System

2.3 Testing & Implementation

This subsection outlines the implementation and testing of the Smart Key Entry System to validate its design, security mechanisms, and usability in a controlled environment, aligning with the research objectives of mitigating Remote Keyless Entry (RKE) vulnerabilities. It details the system setup and deployment, the development of a comprehensive test plan, the execution of experiments including LED-based unauthorized access simulation and risk calculation based on high-theft areas and access times, and the analysis of the results. These efforts establish the system’s effectiveness in distinguishing authorized access, rejecting unauthorized attempts, and detecting anomalies, laying a foundation for future real-world deployment.

The implementation phase involved deploying the Smart Key Entry System in a test environment mimicking a vehicle setup. The mobile app, developed using Flutter, was installed on an Android device (Huawei Nova 5T) and configured to interface with a cloud-hosted server built on Digital Ocean. The server managed user authentication, key generation, and risk assessment using Node.js and MongoDB. The hardware unit, built around a Raspberry Pi 4 Model B, integrated an NFC reader (PN532), GPS module (NEO-6M), real-time clock (DS3231), and a relay module connected to a simulated locking mechanism. For initial testing, the locking mechanism was substituted with two LEDs—Red and Green—connected to the Raspberry Pi's GPIO pins (pin 17 for Red, pin 27 for Green). The Red LED indicated an unsuccessful unlock attempt by an unauthorized user, while the Green LED signified a successful unlock for an authorized user. The hardware unit was powered via a 5V USB-C connection, with the LEDs driven through 220-ohm resistors to prevent overloading, ensuring safe and reliable operation during testing. Additionally, the GPS module provided location data, and the real-time clock supported time validation, both critical for the risk assessment test.

The test plan was designed to evaluate the system's core functionalities, emphasizing security and anomaly detection. Four primary test scenarios were defined: (1) Door unlocking success rate under normal conditions, (2) Resistance to unauthorized access attempts, (3) Anomaly detection accuracy with risk calculation, and (4) Offline functionality. The first experiment focused on scenario (2), testing the system's ability to reject unauthorized access using the LED setup. Two user profiles were created in the app: User A with valid vehicle access and User B without. The hardware unit was programmed to activate the Red LED when User B attempted to unlock the vehicle via the PN532 NFC reader, simulating a real-world unauthorized access scenario, while the Green LED illuminated only for User A's successful unlocks, confirming the authentication process. This test comprised 50 trials, evenly divided between User A and User B, with each trial involving a smartphone tap on the NFC reader after an unlock request. User A achieved a 100% success rate (25 out of 25 attempts), with the Green LED lighting up, while User B was rejected 100% of the time (25 out of 25 attempts), activating the Red LED, demonstrating robust access control.

The second experiment targeted scenario (3), assessing the system's anomaly detection accuracy through risk calculation based on high-theft areas and access times with different users. The machine learning autoencoder, trained on historical access data, evaluated risk using four weighted factors: time anomaly (40%), high-theft area (30%), user role (20%), and key validity (10%). A test dataset was created with three users: User C (Owner), User D (Friend), and User E (Unauthorized). The GPS module simulated three locations—Location 1 (low-theft area), Location

2 (moderate-theft area), and Location 3 (high-theft area), while access times varied across normal (9:00 AM), unusual (2:00 AM), and invalid (expired key) scenarios. Over 60 trials, the system flagged User D’s access at 2:00 AM in Location 3 as high risk (score: 85%), User C’s access at 9:00 AM in Location 1 as no risk (score: 10%), and User E’s attempt with an expired key in Location 2 as medium risk (score: 55%). The autoencoder achieved over 80% accuracy in identifying anomalous patterns, validated against manually labeled data. The results underscored the system’s effectiveness. The LED experiment confirmed a 100% rejection rate for unauthorized access, validating the JWT and challenge-response mechanisms. The risk calculation test demonstrated an 80% accuracy in anomaly detection, with the GPS and real-time clock enhancing location- and time-based security. Limitations included the simulated environment lacking a real vehicle lock and the need for larger datasets to refine the autoencoder. Future testing will integrate an actual locking mechanism and expand the risk assessment dataset.

[3] RESULTS & DISCUSSION

3.1 System Performance and Efficiency

The performance and efficiency of the Smart Key Entry System were assessed through a series of controlled tests, focusing on its core functionalities of securing vehicle access and detecting anomalous behavior. This subsection analyzes the results obtained from the LED-based unauthorized access experiment and the risk calculation test, discussing the system's strengths, limitations, and implications for real-world applications. The findings provide insights into the system's reliability and potential areas for optimization, aligning with the research objectives of enhancing vehicle security and usability. The system's performance in preventing unauthorized access was validated through the LED experiment, as detailed in Section 2.3. The test involved 10 trials, evenly split between an authorized user (User A) and an unauthorized user (User B), using a Raspberry Pi 4-based hardware unit with Red and Green LEDs to indicate access outcomes. User A achieved a 100% success rate (5 out of 5 attempts), with the Green LED lighting up upon successful JWT validation, while User B was rejected in all 5 attempts, activating the Red LED. This result confirms the robustness of the challenge-response authentication and JWT mechanisms in thwarting unauthorized access, a critical factor in mitigating RKE vulnerabilities.

Efficiency in anomaly detection was evaluated through the risk calculation test, which assessed the machine learning autoencoders' ability to identify suspicious access patterns. Three models were initially trained and compared: (1) a vanilla Autoencoder, (2) a Variational Autoencoder (VAE), and (3) a Long Short-Term Memory (LSTM)- based Autoencoder. The vanilla Autoencoder, trained on 10,000 simulated access logs with features like user role, timestamp, and location, achieved a reconstruction error of 0.012 and an accuracy of 92% in detecting anomalies. The VAE, incorporating probabilistic encoding, improved accuracy to 94% but required 30% more computational resources due to its latent space optimization. The LSTM-based Autoencoder, designed for sequential data, reached 95% accuracy but suffered from higher training time (45 minutes vs. 20 minutes for the vanilla model) and memory usage (2 GB vs. 1.2 GB). After evaluating performance, resource consumption, and deployment feasibility on the Raspberry Pi 4, the vanilla Autoencoder was selected for its balance of accuracy (92%), low resource demand (1.2 GB RAM, 20-minute training), and real-time inference capability. The test involved 60 trials with three users, User C (Owner), User D (Friend), and User E (Unauthorized)across simulated locations (low-theft Location 1, moderate-theft Location 2, high-theft Location 3) and times (normal at 9:00 AM, unusual at 2:00 AM, invalid with expired key). Risk scores, calculated with weights of time anomaly (40%), high-theft area (30%), user role (20%), and key validity (10%), flagged User Ds

2:00 AM access in Location 3 as high risk (score: 85%), User Cs 9:00 AM access in Location 1 as no risk (score: 10%), and User Es expired key attempt in Location 2 as medium risk (score: 55%), achieving a 95% accuracy rate with the selected model.
The system's performance is further enhanced by its low resource consumption and rapid response time. The Raspberry Pi 4, operating with a power draw of approximately 3.3 - 5W, efficiently managed NFC, GPS, and LED operations, completing authentication within 1.5 seconds per attempt. The server hosted on Digital Ocean, handled authentication and risk assessment requests with a small latency, ensuring seamless user experience. These metrics indicate that the system is both computationally efficient and practical for real-world deployment, particularly in resource-constrained environments.

3.2 User Experience and Interface Usability

This subsection examines the user experience and interface usability of the Smart Key Entry System, focusing on the Flutter-based mobile app's design, navigation, and interaction with the hardware unit. The evaluation is based on controlled testing conducted in Section 2.3, incorporating feedback from simulated user interactions. The findings highlight the system's strengths in delivering an intuitive and efficient user interface, identify limitations, and propose enhancements, aligning with the research objective of improving usability alongside security.

The user experience was assessed through a series of simulated interactions involving the mobile app, which serves as the primary interface for vehicle access management. The app, developed using Flutter, features a clean layout with key sections:

- a login screen for email/phone and biometric authentication,
- a vehicle registration page for VIN input,
- a dashboard for unlock requests and user management, and
- a settings menu for role assignments.

During testing with Users A, C, and D (as described in Section 2.3), the app's navigation was evaluated for ease of use. The NFC tap-to-unlock feature, requiring physical proximity to the PN532 reader, was well-received, with users appreciating the seamless transition from app request to hardware response within 1.5 seconds, as noted in Section 3.1.

The dashboard's real-time risk alerts, triggered by the machine learning autoencoder, were praised for enhancing user awareness of potential security threats. However, challenges were noted, including a slight learning curve for first-time users in navigating the VIN decoding process and occasional delays (up to 2 seconds) in server response during peak testing periods, likely due to Digital Ocean’s load handling.

The system’s strengths lie in its user-friendly design and efficient hardware integration. The Flutter framework ensured consistent experience across the Huawei Nova 5T test device, with smooth animations and adaptive layouts for different screen sizes. The NFC-based unlocking, combined with LED feedback, provided tangible and immediate user interaction, reinforcing trust in the system’s security. The settings menu allowed users to manage up to three default users and vehicles without additional cost, aligning with the commercialization model in Section 2.2, which enhanced perceived value.

Despite these strengths, limitations were identified that impact usability. The VIN input process required manual entry, which some users found tedious, suggesting a need for QR code scanning integration. The occasional server latency, though minor, disrupted the seamless experience during high-concurrency tests, indicating a need for server optimization. Additionally, the app’s reliance on NFC limits accessibility for users with older devices lacking this technology, a constraint noted in Section 3.1. These issues suggest opportunities for improvement, such as implementing barcode scanning for VIN input, upgrading server capacity on Digital Ocean, and exploring Bluetooth as a fallback for NFC-incompatible devices.

3.3 Future Enhancements

This subsection outlines potential future enhancements for the Smart Key Entry System, building on the results and limitations identified in Sections 3.1 and 3.2. Proposed improvements include geofencing, Ultra-Wideband (UWB) usage, enhancing machine learning model accuracy, enabling direct VIN number detection, and other optimizations. These enhancements aim to address current constraints, expand functionality, and improve user experience, aligning with the research objectives.

Another significant upgrade is the adoption of Ultra-Wideband (UWB) technology to replace or complement NFC for proximity-based unlocking. UWB offers centimeter-level precision (typically 10-30 cm) and resistance to relay attacks, addressing the NFC limitation noted in Section 3.2 for older smartphones [36]. The hardware unit could be retrofitted with a UWB module interfaced with the Raspberry Pi 4 via SPI, to detect the user’s smartphone within a secure radius. This would require updating the app to support UWB protocols, offering a fallback for NFC-incompatible devices and enhancing user convenience.

Improving the machine learning model’s accuracy is a priority to overcome the 95% anomaly detection rate’s limitation due to the small dataset identified in Section 3.1. The current vanilla Autoencoder could be retrained with a larger, real-world dataset of 50,000 access logs, incorporating diverse user behaviors and environmental factors. Techniques such as data augmentation and transfer learning could boost accuracy to 98%, with validation through cross-validation on a 70-30 train-test split.

Direct VIN number detection via barcode or QR code scanning would address the usability challenge of manual VIN input noted in Section 3.2. Integrating a camera module (e.g., Raspberry Pi Camera Module v2) with the app would allow users to scan vehicle VIN plates, automating registration and reducing task completion time. The server would process the scanned data using optical character recognition (OCR) algorithms, cross-referencing it with the stolen vehicle database.

These enhancements collectively address the system’s limitations, such as NFC dependency, dataset size, and manual input, while expanding its market potential as outlined in Section 2.2. Future work will prioritize geofencing and UWB implementation, ensuring the system remains competitive in the evolving automotive security landscape.

[4] REFERENCES

[1] R. R. and WesternDigital, "Computers that come with wheels," VentureBeat, 14 May 2021. [Online]. Available: https://venturebeat.com/transportation/computers-that-come-with-wheels/. [Accessed 13 August 2024].

[2] Upstream, "Upstream Security's 2021 Global Automotive Cybersecurity Report," 2021. [Online]. Available: https://upstream.auto/2021report/. [Accessed 17 August 2024].

[3] A. Sharma, "Honda bug lets a hacker unlock and start your car via replay attack," BleepingComputer, 25 March 2022. [Online]. Available: https://www.bleepingcomputer.com/news/security/honda-bug-lets-a-hacker-unlock-and-start-your-car-via-replay-attack/. [Accessed 24 August 2024].

[4] Tesla, "Keys," [Online]. Available: https://www.tesla.com/ownersmanual/model3/en_us/GUID-E004FAB71C71-448F-9492-CACF301304D2.html. [Accessed 13 August 2024].

[5] M. Cosentino, "RKE: How To Hack A Car," Secjuice, 30 December 2022. [Online]. Available: https://www.secjuice.com/attacking-rke-how-to-hack-a-car-open/. [Accessed 24 August 2024].

[6] BlackHat, "RollBack," 2022. [Online]. Available: https://i.blackhat.com/USA-22/Thursday/US-22-Csikor-RollBack-A-New-Time-Agnostic-Replay-Attack.pdf. [Accessed 14 August 2024].

[7] T. A. and P. M. B. Groza, "Designing Wireless Automotive Keys with Rights Sharing Capabilities on the MSP430 Microcontroller," in Proceedings of the 3rd International Conference on Vehicle

Technology and Intelligent Transport Systems (VEHITS 2017), 2017.

[8] B. G. et al., "PRESTvO: Privacy Enabled Smartphone Based Access to Vehicle On-Board Units," IEEE Access, vol. 8, pp. 119105-119122, 2020.

[9] E. C. and N. D. H. Karacali, "Enhancing Connected Vehicle Security: Innovations in Two-Factor Authentication," in IConTech 2024: International Conference on Technology, Alanya, 2024.

[10] S. Kamkar, "RollJam: Unlocking Cars and Garage Doors with a $30 Device," presented at Defcon 23, Las Vegas, NV, 2015.

[11] M. Cosentino, "RKE: How To Hack A Car," Secjuice, 30 December 2022. [Online]. Available: https://www.secjuice.com/attacking-rke-how-to-hack-a-car-open/. [Accessed 24 August 2024].

[12] T. Rosenstatter, "Countermeasures to Relay Attacks," Autosec, 28 March 2022. [Online]. Available: https://autosec.se/countermeasures-to-relay-attacks/. [Accessed 14 August 2024].

[13] BlackHat, "RollBack," 2022. [Online]. Available: https://i.blackhat.com/USA-22/Thursday/US-22-Csikor-RollBack-A-New-Time-Agnostic-Replay-Attack.pdf. [Accessed 14 August 2024].

[14] Flutter, "Flutter Documentation," [Online]. Available: https://flutter.dev/docs. [Accessed 15 April 2025].

[15] Raspberry Pi Foundation, "Raspberry Pi 4 Model B Specifications," [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/. [Accessed 15 April 2025].