SCHOOL OF COMPUTING, TECHNOLOGY & APPLIED SCIENCES

Computing Project

A Security aware IoT-Driven Remote Patient Monitoring in Health Care Framework

Names: Chakamba Virginia

Student Number: BSc12020

Programme: Master of Science in Computer Science

Supervisor: Prof. Aaron Zimba

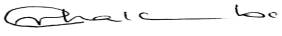
DECLARATION

Name: Viriginia Chakamba

Student Number: BSc12020

I hereby declare that this final year research project is the result of my own work, except for quotations and summaries which have been duly acknowledged.

Plagiarism check: %

Signature:  Date: 30/06/2025

Supervisor Name:

Supervisor Signature:

Date:

ABSTRACT

The increasing adoption of the Internet of Things (IoT) in healthcare has revolutionized patient care through real-time monitoring and remote diagnostics. This project presents a Security-Aware IoT-Driven Remote Patient Monitoring (RPM) Framework that captures, encrypts, and securely transmits vital patient data over the internet. Built on the ESP32 microcontroller, the system integrates multiple biomedical sensors—including DHT22 for temperature, pulse sensors for heart rate, analog sensors for SpO_2, and blood pressure sensors—to continuously collect physiological parameters.

To safeguard sensitive health data during transmission, the framework employs a hybrid cryptographic approach, combining Advanced Encryption Standard (AES) for fast symmetric encryption and Elliptic Curve Cryptography (ECC) for secure key exchange and asymmetric encryption of the AES key. This encrypted data is transmitted over the MQTT protocol using TLS to a secure cloud broker (HiveMQ), ensuring data confidentiality and integrity during communication.

On the cloud end, the system uses a FlowFuse-hosted Node-RED environment to visualize decrypted sensor data on a real-time dashboard. Node-RED workflows are designed to handle decryption, verification, and display, allowing authorized healthcare providers to access the data in human-readable form. This layered architecture addresses critical security vulnerabilities such as unauthorized access, data interception, and tampering.

The proposed framework is scalable, low-power, and cost-effective, making it ideal for deployment in home care, rural clinics, and telehealth applications. It demonstrates a practical balance between real-time performance and high-level data security, marking a significant contribution to secure IoT healthcare solutions.

Keywords:

Remote Patient Monitoring (RPM), Internet of Things (IoT), Advanced Encryption Standard (AES), Elliptic Curve Cryptography (ECC), MQTT, TLS, Node-RED, FlowFuse, ESP32, Health IoT, Secure Communication.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who have supported and guided me throughout the development of this project.

First and foremost, I thank **God Almighty** for granting me the strength, perseverance, and clarity to complete this work.

I extend my heartfelt appreciation to my **supervisor** Professor Aaron Zimba whose valuable insights, encouragement, and continuous support were instrumental in shaping this project. Your constructive feedback and expert guidance helped me stay focused and motivated.

Special thanks go to the **faculty and staff** of School of Computing, for providing the technical foundation and facilities necessary to carry out this research.

Lastly, I would like to acknowledge the unwavering support of my **family and friends**, whose encouragement and patience helped me overcome every challenge during this journey.

THANK YOU.

DEDICATION

This project is **dedicated** to my beloved **family**, whose unwavering love, prayers, and encouragement have been a constant source of strength throughout my academic journey.

To my **lecturers and mentors**, who inspired my interest in technology and healthcare innovation, I am deeply grateful for your guidance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION	MEANING
IoT	Internet of Things
RPM	Remote Patient Monitoring
ECC	Elliptic Curve Cryptography
AES	Advanced Encryption Standard
MQTT	Message Queuing Telemetry Transport
MQTT-SN	MQTT for Sensor Network
TLS	Transport Layer Security
WIFI	Wireless Fidelity
ESP32	Espressif Systems Microcontroller
DHT22	Digital Humidity and Temperature Sensor
SPO2	Peripheral Capillary Oxygen Saturation
HR	Heart Rate
BP	Blood Pressure
UI	User Interface
GUI	Graphical User Interface
API	Application Programming Interface
JSON	JavaScript Object Notation
CA Cert	Certificate Authority Certificate
LED	Light Emitting Diode
IP	Internet Protocol
SSL	Secure Socket Layer
PKI	Public Key Infrastructure

CHAPTER 1

INTRODUCTION

1.1 Background to the study

With the goal of keeping patients as comfortable as possible during their everyday lives most healthcare facilities are now using Remote Patient Monitoring (RPM) systems.

RPM systems use digital technologies to collect health data from individuals in one location, such as a patient's home, and electronically transmit the information to healthcare providers in a different location for assessment and recommendations[1].

This makes it possible for some patients to do away with in-person visits with their provider. Instead, these patients may be cared for either at their home/place of residence or, as required, at any other non-hospital location for both acute and chronic conditions[2].

Due to advancements in technology and the increasing demand for efficient patient-centred care, the healthcare industry is undergoing a major transformation. One of the most promising developments in this field is the integration of the Internet of Things (IoT) into Remote Patient Monitoring (RPM).

Uddin and Koo noted that IoT-driven RPM systems utilize interconnected devices and sensors to collect, transmit, and analyse health data in real-time, providing continuous monitoring of patients outside traditional clinical settings.[3]

While RPM are aimed at improving the welfare of patients, it has several challenges including data security and privacy, device accuracy, connectivity issues, patient engagement and clinical integration [4].

This research explores the current state of IoT-driven RPM, existing frameworks, and key challenges, it will take keen interest on the aspect of security.

1.2 Problem Statement

The increasing implementation of Internet of Things (IoT) technologies in healthcare has transformed Remote Patient Monitoring (RPM), offering real-time data collection and facilitating timely interventions[4].

Conversely, as the use of IoT in health gains speed, the risk to the security of sensitive health data in transit and at rest has become apparent. Many IoT devices in healthcare, such as wearables and sensors, do not possess formidable security features, leaving patient data vulnerable to cyber-attacks, unauthorized access, and data breaches [5].

The security of IoT-based Remote Patient Monitoring (RPM) systems is critical to ensuring the protection of sensitive health data and the safety of patients.

However, several issues exist with current security technologies employed in these systems, which hinder their effectiveness. The security issues that standout in the area include:

1. Weak encryption.
   [6] alludes that most wearable sensor devices have limited computing power and therefore, cannot perform traditional cryptographic calculations
2. Insufficient Authentication and Access Control as most authentication protocols have limitation[7].
3. Lack of standardised security protocols that leads inconsistent security measures due to variations in security implementation across devices [9].

Therefore, there is a critical need for a Security-Aware IoT-Driven Remote Patient Monitoring Framework that prioritizes the protection of patient data through available security technologies. This framework should integrate security at every level from the collection and transmission of data to its analysis and storage. The goal is to build a secure, scalable, and efficient IoT-driven RPM system that improves patient care while safeguarding sensitive health information from potential threats.

1.3 Aim

This research aims to develop a Security Aware Internet of Things (IoT) driven Remote Patient Monitoring (RPM) framework that will help improve the security of patient data in transit and at rest.

1.4 Objectives of the study

The main objectives of the study are:

1. Investigate the landscape of IoT based frameworks for Remote Patient Monitoring to ascertain the challenges that exist.
2. Develop an IoT driven framework for remote patient monitoring.
3. Implement appropriate security mechanisms for secure data transmission between devices.
4. Evaluate the performance of the framework in relation to secure transmission of data using corresponding IoT security technologies.

RESEARCH QUESTIONS

1. What are the primary challenges and significant security vulnerabilities associated with the integration of IoT-driven frameworks in Remote Patient Monitoring systems?
2. How can one design a security aware IoT-driven framework for remote patient monitoring which can seamlessly be integrated with existing healthcare systems?
3. How effective are the current security technologies implemented in the existing frameworks in ensuring the secure transmission of data, and what is their impact on system performance?

1.5 Scope and Limitation

This research aims to investigate, design, and evaluate an IoT-based framework for Remote Patient Monitoring (RPM), paying particular attention to addressing the security concerns associated with the transmission of patient's health data. The study will explore the current state of IoT frameworks, develop a new secure framework, and assess its performance in ensuring secure data transmission.

The study will investigate existing IoT frameworks in healthcare settings and identify the challenges associated with their application in remote patient monitoring. The challenges will include data security, device interoperability, network reliability, and integration with healthcare systems. The study will however focus on assessing the security measures currently implemented in these frameworks (e.g., encryption, authentication) and highlight gaps that could be improved [10].

Based on the identified challenges, the research will develop a secure IoT-driven framework for RPM that will include the integration of IoT devices with secure data transmission protocols while addressing key security issues such as encryption techniques, and secure authentication and authorization mechanisms [11].

The framework will then be tested to evaluate its performance in terms of secure data transmission focusing on key indicators such as strength of encryption algorithms, latency and real-time data transfer. It will also assess how the framework handles security challenges during data exchange between IoT devices and healthcare providers, ensuring the confidentiality and integrity of patient health data [12].

However, the research will focus on securing data during transmission and not address other security aspects such as device-level security or protection against physical tampering.

It will focus on research journal articles published in the last 5 years.

1.6 Significant of the Project

This study is of great importance as it addresses the emerging role of the Internet of Things (IoT) in enhancing healthcare delivery, particularly around Remote Patient Monitoring (RPM).

This significance lies in its ability to improve healthcare delivery through IoT technologies while addressing critical security concerns that currently hinder the effectiveness of Remote Patient Monitoring systems.

By investigating current frameworks, developing a secure solution, and evaluating its real-world performance, this research will pave way for safer, more efficient, and accessible healthcare systems that leverage the potential of IoT to provide better patient care [13].

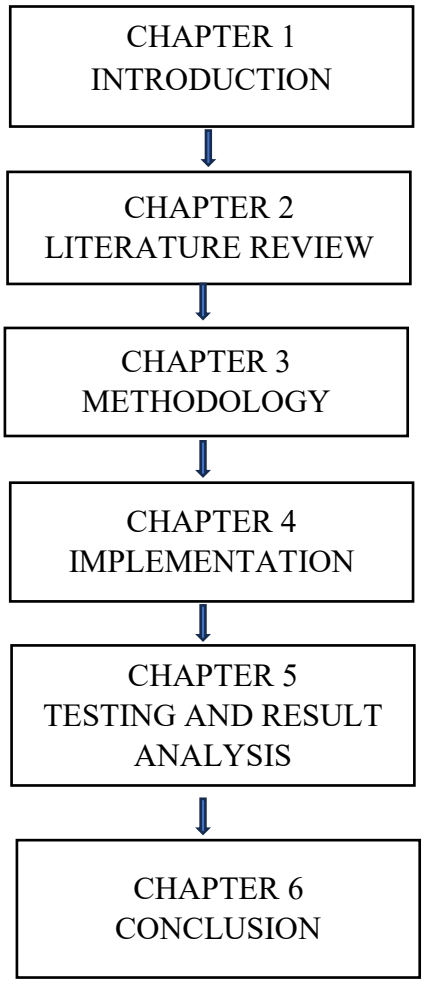
1.7 Chapter 1 - Summary

The chapter introduces the need for a Security-Aware IoT-Driven Remote Patient Monitoring (RPM). It provides background information, defines the problem, outlining objectives. It also highlights the significance of the research. It discusses the growing implementation of Remote Patient Monitoring (RPM) systems in healthcare, which takes advantage of IoT technology to collect and transmit patient data for remote assessment. It highlights key security challenges in

IoT-based RPM. The challenges include weak encryption, inadequate authentication mechanisms, and lack of standardized security protocols.

1.8 Preliminary sections of the project report

The chart below outlines the outlines of the report:



CHAPTER 2

LITERATURE REVIEW

2.1 General Background

Remote Patient Monitoring (RPM) is new healthcare provision method that uses digital technologies to collect and electronically transmit health data from individuals in one location to healthcare providers in another for assessment and recommendations[8]. The integration of the Internet of Things (IoT) into healthcare has revolutionized Remote Patient Monitoring (RPM). IoT refers to a network of interconnected devices equipped with sensors, software, and other technologies that facilitate the collection and exchange of data over the interne [14] and has many application areas such as agriculture, Smart Cities, Industrial Automation etc. [9]. In healthcare, the technology has meaningfully enhanced RPM by enabling real-time, continuous, and automated monitoring through interconnected devices such as wearable sensors, smart implants, and home-based health equipment. However, the adoption of IoT in Remote Patient Monitoring (RPM) has also brought several challenges and security concerns stands out prominent among them[10].

2.2 Broad Literature review of the topic

Remote Patient Monitoring (RPM) has stood out new paradigm shift approach in healthcare that enables continuous tracking of patients' health metrics away from traditional clinical settings. This has become inevitable because the rising prevalence of chronic diseases and an aging population are placing increasing pressure on healthcare systems[8]. Smart wearable accessories and implantable, such as smartwatches, bracelets, armbands[11], are more and more becoming part of everyday life. These devices when worn on the body can monitor, analyse, and manage health conditions.

Remote Patient Monitoring (RPM) has significantly transformed by the integration of the Internet of Things (IoT) technology[8]. The Internet of Things (IoT) is a new model and most certainly one of the most important technological revolutions. IoT is a standard that involves physical intelligent objects capable of sensing, processing information and communication through wireless or wired connection. The devices are capable making decision thus making them smart things[12].

Enabled with IoT technology, RPM systems can transmit continuous, real-time patients' health metrics[13] enhancing proactive care and reducing the need for frequent hospital visits.

Key components of IoT driven Remote Patient monitoring are:

1. IoT Devices and Sensors that are used to collect vital signs such as heart rate, blood pressure, and glucose levels enabling continuous health monitoring and allowing for timely medical interventions. Such devices include smartwatches, ECG monitors and implantable sensors[14].
2. Connectivity and Data Transmission the data that has been collected by the IoT devices is transmitted via wireless networks (e.g., Wi-Fi, Bluetooth)[15] to centralized systems or cloud platforms for processing and storage insuring seamless data flow between patients and healthcare providers.
3. Data Storage and Analysis technologies such cloud and edge computing are used to store and process patient data transmitted by IoT devices
4. Integration with Healthcare Systems: the IoT-based is the integrate with the local health care systems providing healthcare professional comprehensive patient data to act on.

2.3 Critical review of related works

The integration of IoT into Remote Patient Monitoring has significantly enhanced healthcare delivery by enabling continuous, real-time monitoring, improving patient outcomes, while reducing healthcare costs. However, this advancement brings forth significant security challenges due to the sensitive nature of health data and the resource-constrained environments of IoT devices. Ongoing research and technological advancements continue to address challenges related to data security, interoperability, and system scalability, paving the way for more robust and efficient RPM solutions.

The reviewed literature outlines several security risks that are apparent when using IoT in healthcare delivery.

1. Weak Encryption

   Encryption is the process of scrambling data into unreadable data using a special algorithm and a key with a goal of protecting the information so that only someone with the correct key can turn it back into its original form (a process called decryption). This keeps sensitive information private and secure, when it's being sent over the internet and stored. [16] highlights how the use of outdated encryption algorithms threatens the confidentiality and integrity of patient data while also outlining limited encryption resources and short key

lengths as being some vulnerabilities posed on the data. [17] evaluated several lightweight cryptographic algorithms used in IoT devices such PRIDE, IDEA ITUbee and rated them as good performers. However, most of them have had significant attacks. Another literature in[18] discussed the use of Advanced Encryption Standard (AES) with a 256 bit key to secure data. The system uses Electronic Codebook (ECB) cipher mode with ZERO padding for encryption and decryption processes. While it protects against various attacks such as brute force, man-in-the middle attacks etc, it is susceptible to pattern recognition and requires substantial computational resources that can overwhelm IoT devices. [19] discusses the use Elliptic Curve Cryptography (ECC) and collision-resistant one-way hash function as an efficient way of securing data on IoT based system. [20].

[21] proposed a framework that pair the lightweight messaging protocol MQTT with TLS encryption to secure patient in transit. However, TLS requires a lot of memory and computational power which most tiny IoT devices do not possess. It also only encrypts data in transit but the data at rest.

2. Authentication and Access Control

These two being fundamental principles in data security work together to ensure only authorised entities have access to the resources. It prevents unauthorised individuals from masquerading and access restricted information.

[22] proposed an authentication process using a combination of Length Ceaser Cipher-based Pearson Hashing Algorithm (LCC-PHA), ECC, and a Fisher-Yates shuffled Adelson-Velskii and Landis (FYS-AVL) tree structure. The mechanism's aim was to prevent unauthorized access and ensure data integrity. However, the framework remains vulnerable to internal attacks. Access control is a key aspect of securing patient data, ensuring that information is only accessed by authorized individuals. [23] proposed a light weight and privacy-preserving mutual authentication protocol that uses cuckoo filters for remote patient monitoring (RPM) systems. Its focus is to reduce computational costs but still preserving strong authentication and privacy. However, the scheme lacks scalability as it was tested on smaller networks and is likely to fail in a real hospital setting. Access control is equally important in Remote Patient Monitoring. [24] suggested an Attribute-Based Access Control (ABAC) that offers fine-grained access control by evaluating attributes of users and devices. This mechanism is effective, but its centralized architecture can lead to complexity and scalability issues in dynamic IoT environments.

2.4 Comparison with related works

Table 2.4.1: Table of Comparison

No.	Security Domain	Security Mechanism	Strengths	Limitation	Gaps identified
1	Encryption	Outdated algorithms (e.g., DES, 3DES) [11]	Low resource consumption	Weak against modern attacks	Fails to ensure data confidentiality and integrity
2	Encryption	Lightweight algorithms (e.g., PRIDE, ITUbee) [12]	Designed for IoT; efficient for constrained devices	Proven vulnerable to cryptanalysis	Needs stronger, more resilient lightweight alternatives
3	Encryption	AES-256 (ECB mode, ZERO padding) [13]	High resistance to brute force, man-in-the-middle attacks	ECB leaks data patterns; high computational demands	Incompatible with low-power RPM devices
4	Encryption	ECC with one-way hash functions [14], [15]	Short key length with strong security	Implementation complexity	Requires simplification for lightweight deployment
5	Encryption + Communication	MQTT with TLS [16]	Secures data in transit; standard in IoT communication	TLS is resource-heavy; no protection for data at rest	Inefficient for low-memory RPM devices
6	Authentication	LCC-PHA + ECC + FYS-AVL Tree [17]	Combines cryptographic integrity and fast data structure traversal	High internal system complexity	Remains susceptible to insider threats
7	Authentication	Cuckoo-filter-based mutual authentication [18]	Low computational load; privacy-preserving	Not tested in large networks	Scalability is unproven in real hospital systems
8	Access Control	Attribute-Based Access Control (ABAC) [19]	Fine-grained policy enforcement based on	Centralized design increases complexity	Not scalable for large-scale IoT deployments

			user/device attributes		
9	Access Control	Role-Based Access Control (RBAC) with temporal policies [20]	Simple to implement and manage; well-understood model	Lacks context-awareness; roles become rigid	Poor adaptability to dynamic IoT environments
10	Access Control	Usage Control (UCON) models integrated with	Continuous access evaluation; flexible with obligations and conditions	High resource requirement and complex policy management	Not yet optimized for heterogeneous healthcare IoT systems

2.5 Identified Gaps

Several gaps have identified by this research as follow:

1. Most security solutions are not optimized for constrained IoT environments such as RPM systems.
2. Most solutions lack real-world validation for security mechanisms in large or dynamic healthcare environments.
3. Some solutions offer incomplete data protection due to use of weak or partial security schemes (e.g., no encryption at rest, vulnerable cipher modes).

2.6 Conceptual framework

The conceptual framework for a security-aware IoT-driven Remote Patient Monitoring (RPM) system will comprise several integrated components, ranging from data collection to data storage and analysis. The components will work together to provide patients with continuous, secure, and real-time health monitoring, putting in mind the confidentiality, integrity, and availability requirements for the sensitive medical data.

Key Components:

1. Data Acquisition and Device Integration

   The framework will support a wide range of IoT sensors what will be used to collect health data from patients. These will include Physiological sensors such as wearables (smartwatches, wristbands, fitness trackers) and implantable (pacemaker, glucometers) [25]that will continuously collect data such as heart rate, blood pressure, oxygen saturation

levels, blood glucose levels, temperature and other vital signs necessary for health monitoring.

2. Secure Data Transmission and Storage

   Secure data transmission and storage are critical to maintaining the confidentiality, integrity, and availability of patient health data in IoT-based Remote Patient Monitoring. It is therefore essential to implement robust security mechanisms to protect data at every stage (from collection and transmission to storage and analysis).

   Key elements to secure data include end-to-end encryption using various encryption schemes, Authentication where devices must authenticate themselves to the network and other devices to prevent unauthorized access using different authentication protocols, secure communication using reliable communication protocols such as MQTT, CoAP and security of data at rest[26].

3. Data analysis and Decision making

   In IoT-driven Remote Patient Monitoring (RPM) systems, data analysis and decision-making are crucial components that transform raw sensor data into actionable medical insights[27]. These processes allow healthcare providers to make timely, accurate, and personalized decisions regarding patient care.

   Data received from sensors will be pre-processed to remove noise or outliers from sensor devices. Then the data will be normalised to bring it to a consistent state and later aggregated over time intervals for trend analysis.

   The figure below depicts the components of the framework.

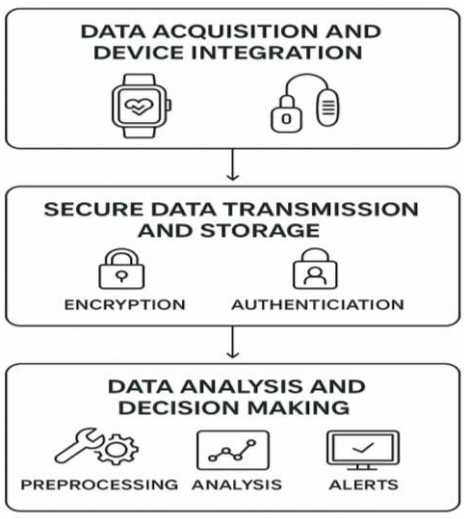


Figure 2.1: Proposed IoT Framework for Remote Patient Monitoring

2.7 Proposed model/system

The focus of this model is to provide a security-aware framework for IoT driven Remote Patient Monitoring (RPM). Its aim to integrate novel and lightweight security protocols that ensure the confidentiality, integrity, and availability (CIA)[28] of health data and system resources thereby addressing the unique challenges posed by resource-constrained IoT devices and the sensitive nature of patient information. This will offer a strong foundation for secure, continuous, and real-time patient monitoring by implementing end-to-end encryption, robust authentication and access control measures, including secure mechanisms for data transmission and storage.

The system architecture will include the following components:

1. Data Acquisition Layer

   This is where data will be collected from patients using IoT sensors, wearable devices, implantable.

   The following security features will be used

   - Elliptic Curve Cryptography (ECC) a public key encryption method that is highly suitable for lightweight encryption at the device level in IoT-based Remote Patient Monitoring (RPM)[29] systems with the ability to provide strong security with significantly smaller key sizes and lower computational overhead compared to traditional encryption schemes
   - Secure Boot & Firmware Validation which is a security mechanism designed to prevent compromised or unauthorized IoT devices from joining the network.

2. Secure Communication Layer

   This is the layer responsible for insuring confidentiality, integrity and availability of data being transmitted between IoT devices to gateways, gateways to the cloud for storage and from the cloud to healthcare providers.

   The Message Queuing Telemetry Transport for Sensor Network (MQTT-SN) protocol combined with Datagram Transport Layer Security (DTLS) which is a lightweight protocol ideal for Wireless Sensor Networks and constrained devices used to secure data in transit[30]. These will provide end to end encryption and mutual authentication between devices

3. Authentication and Access Control Layer

   Only authorized devices and users are permitted access to the RPM system, and the Authentication and Access Control Layer guarantees that this requirement is met.

   For Authentication, a multi-factor authentication combining biometric data and digital certificates will be used while an Attribute-Based Access Control (ABAC) providing fine-grained, policy-driven permissions will be used for access control[31].

4. Data Storage and Management Layer

   This layer is deals with secure, efficient, and scalable storage of patient health data collected by IoT devices. It ensures that sensitive information remains protected both during and after storage. AES-256 encryption with Cipher Block Chaining (CBC) or Galois/Counter Mode (GCM) instead of ECB to avoid pattern vulnerabilities.

2.8 Chapter Summary

This chapter provides a comprehensive overview of Remote Patient Monitoring (RPM), emphasizing its evolution through the integration of Internet of Things (IoT) technologies. It outlines how IoT enhances healthcare delivery by enabling real-time, continuous monitoring using smart wearables, sensors, and connected devices. The broad literature review explores RPM's significance in managing chronic conditions and supporting an aging population, and highlights key components such as data acquisition, transmission, storage, and system integration. The critical review identifies major security challenges, particularly weak encryption, inadequate authentication, and access control mechanisms. Various security approaches are assessed, including lightweight cryptographic algorithms, mutual authentication protocols, and fine-grained access control models, with comparisons showing existing gaps in scalability, complexity, and resource efficiency. A conceptual framework is then proposed, focusing on secure data acquisition, transmission, and storage, as well as intelligent data analysis for decision-making. Finally, a security-aware model is introduced that leverages ECC for lightweight encryption, secure communication via MQTT-SN and DTLS, multi-factor authentication, and ABAC-based access control, supported by robust encryption methods like AES-CBC and AES-GCM to ensure the confidentiality, integrity, and availability of patient data in resource-constrained IoT environments.

CHAPTER 3

METHODOLOGY

This chapter aims to describe the methodology that will be used to develop the framework for Remote Patient Monitoring with security as the focus. It will describe the main steps and processes need to come up with a coherent framework that ensure the security of patient during transmission and at rest

3.1 Research design

Design Science Research method was adopted for this because it provides a clear, step-by-step methodology from problem identification, objective definition, design, development, demonstration, evaluation, and communication. The research identifies a real-world healthcare problem, that is, the lack of secure and integrated RPM systems. The Design Science Research (DSR) methodology is chosen for this study because it is ideally suited for addressing practical and technological challenges through the design, development, and evaluation of innovative artifacts[32].

In the context of a security-aware IoT-driven Remote Patient Monitoring (RPM) system, the main objective goes beyond simply analyzing existing issues but focuses on designing and implementing a practical and secure solution that effectively meets real-world healthcare demands. [32] describes Design Science Research principles as a systematic development of IS artifacts that are both innovative and grounded in scientific research, facilitating cumulative knowledge building in the field. The use of DSR aims to reduce the gap between theoretical research and practical implementation, ensuring that digital innovations are both scientifically rigorous and practically relevant

Due to its emphasis on a problem-solving development method, from identifying the problem to artifact design, demonstration, evaluation, and refinement, Design Science Research fits seamlessly in the design of a Security-aware framework for an IoT-driven Remote Patient Monitoring.

Below is the design science research process diagram showing the stages involved:

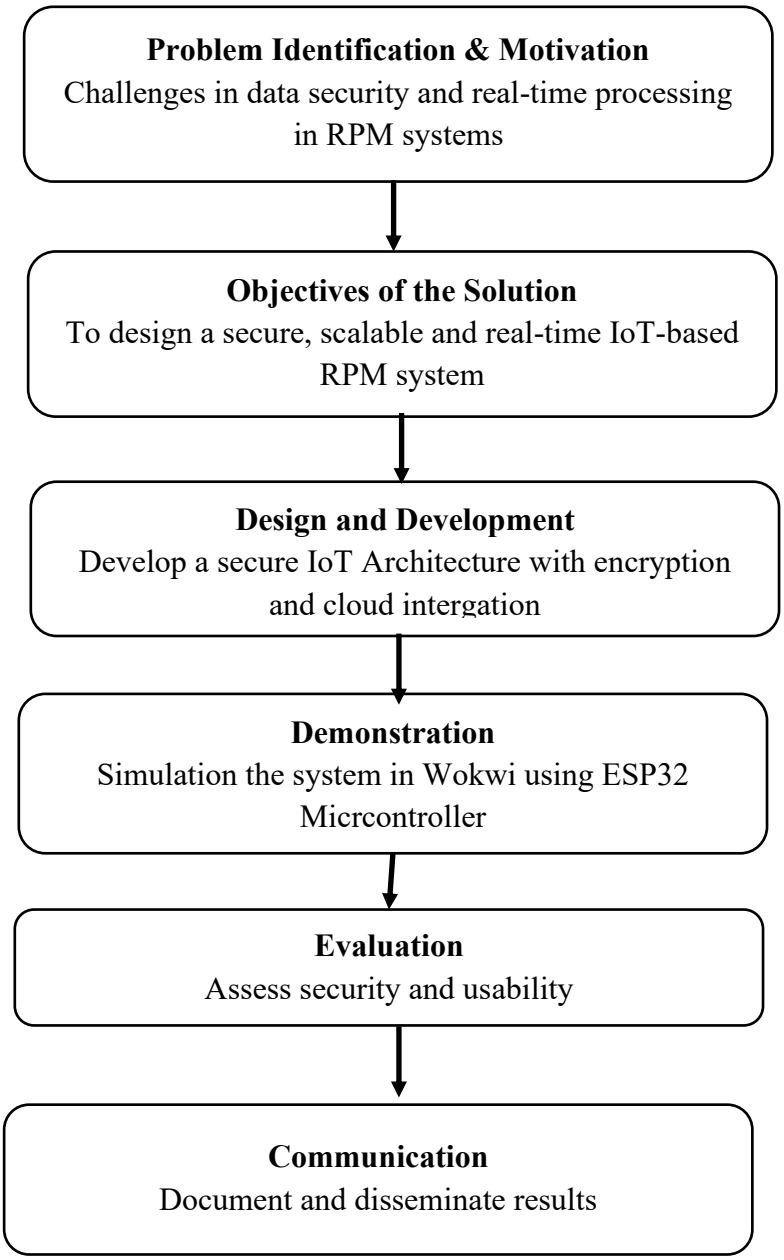


Figure 3.2: Proposed IoT Framework for Remote Patient Monitoring

The table below outlines the steps involved the Design Science Research methodology in relation to a Security-aware framework for IoT-driven Remote Patient Monitoring.

Table 3.1: DSR for IoT driven RPM

Security-aware framework for IoT driven Remote Patient Monitoring	
DSR Component	Description
Problem Identification	Traditional RPM systems face risks related to data security
Design Objectives	1. Investigate the landscape of IoT based frameworks for Remote Patient Monitoring to ascertain the challenges that exist. 2. Develop an IoT driven framework for remote patient monitoring. 3. Implement appropriate security mechanisms for secure data transmission between devices. 4. Evaluate the performance of the framework in relation to secure transmission of data using corresponding IoT security technologies
Design & Development	Design and develop a multi-layer IoT-based RPM framework comprising: • IoT sensors (wearable, implantable) • Wireless communication • AES/ECC-based encryption • Authentication & routing • Cloud for storage/analytics
Artifact	A working prototype simulated in Wokwi using ESP32 board as a microcontroller demonstrating secure RPM data collection, secure transmission, and cloud-based storage.
Demonstration	• Simulate IoT data flow using ESP32 • Show data encryption and forwarding through gateway • Visualize network activity and latency
Evaluation	Evaluate the system based on: • Security (encryption success, threat resistance) • Performance (latency, packet delivery ratio)
Communication	Share through: • Technical documentation • Simulation visualizations

3.2 Methodological Framework

This research aims to develop a framework for Remote Patient Monitoring, prioritising the security of patient data as it is being transmitted and stored. The methodological framework will have several elements which will include data collection using various sensor devices, secure data transmission and secure storage. The flow chart below shows the steps involved in the research.

3.3 Association of research method to project

Developing a security-aware Internet of Things (IoT)-driven Remote Patient Monitoring (RPM) system requires rigorous design validation, especially given the sensitivity of the data in question and the criticality of timely response. In this context, experimental prototyping using simulation tools such as Wokwi and ESP32 microcontroller provides a practical and low-risk approach to evaluate the system's behaviour under varying network and security configurations prior to real-world deployment. The system under study involves collecting physiological data such as blood pressure, respiratory rate, heartbeat, or other health indicators from wireless sensors attached to or implanted in the patient[33]. These sensors communicate through a wireless medium, either directly or via a gateway, to a remote healthcare server that processes and stores the data. Due to the potentially vulnerable nature of wireless communication, it is crucial to integrate lightweight cryptographic techniques, such as Elliptic Curve Cryptography (ECC) and Advanced Encryption Standard (AES), to secure patient information while maintaining efficiency and low power consumption.

Experimental prototyping in simulated environments enables the step-by-step demonstrating of this system, beginning with the basic communication architecture i.e. sensors, access points, gateways, and cloud nodes after which additional features such as encryption, authentication, and mobility can be added. Wokwi and ESP32 facilitate this by offering modular components and customizable protocol stacks[34].

Furthermore, the simulation enables the testing of various network topologies and attack scenarios.

Ultimately, the experimental prototyping through simulation not only accelerates development but also ensures that the RPM system meets the stringent requirements of healthcare applications in terms of security, reliability, and real-time responsiveness approach and lays the groundwork for future hardware implementation, offering developers and researchers a robust

framework to test, refine, and validate their systems in a controlled, reproducible, and scalable manner.

Apart from the above, simulation and experimental prototyping provide inexpensive, efficient and scientifically valid approach to design, experiment, and enhance the proposed IoT system.

Below is the framework architecture showing all the essential components required to achive the objectives.

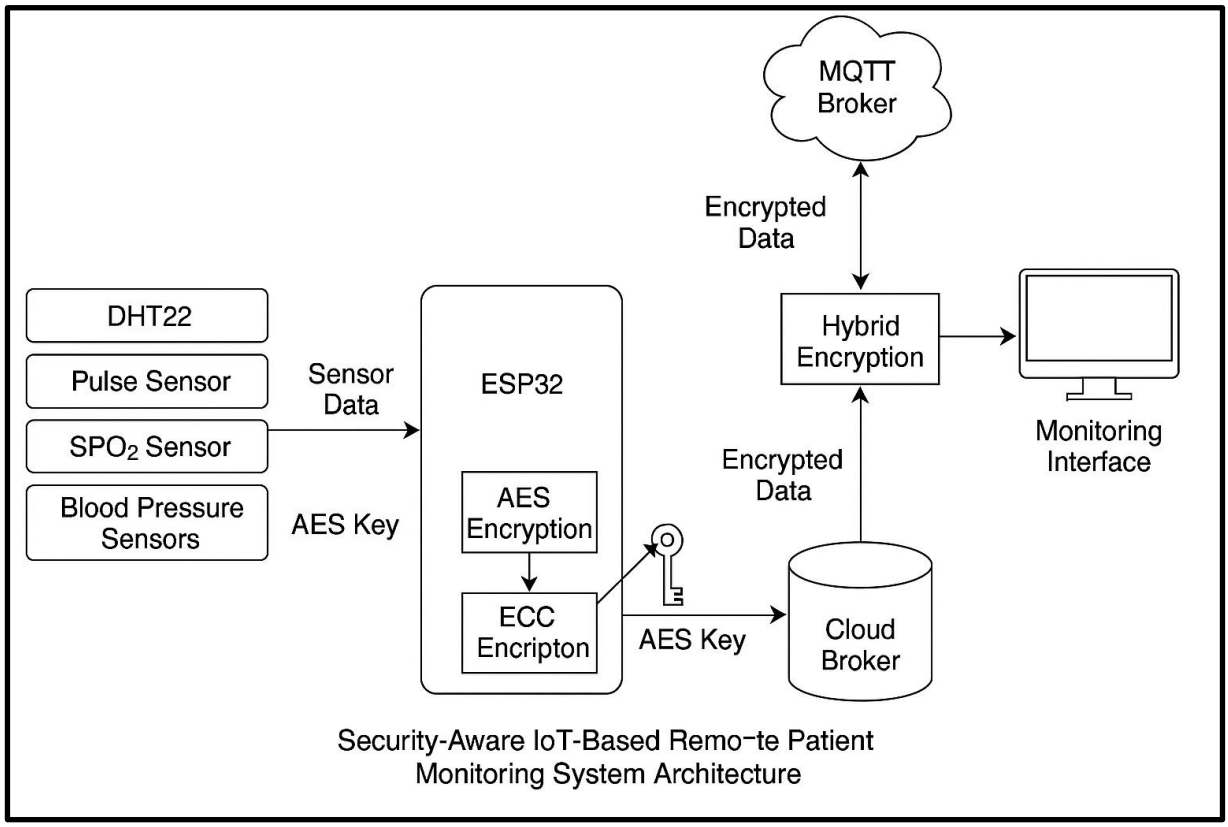


Figure 3.3: Framework Architecture

3.4 Research data and datasets

In this study, the simulation of a security-aware IoT-driven Remote Patient Monitoring (RPM) system will be carried out using the Wokwi simulation environment together with the ESP32 microcontroller. Instead of relying solely on externally collected data, the simulation will generate comprehensive datasets that represent network behavior, security processing, and system performance. These datasets provide insight into the operational characteristics of the RPM system under both normal and security-enhanced conditions.

3.4.1 **Data types to be used**

In this remote patient monitoring system, various medical sensors will be employed to continuously track vital health parameters. Each of these sensors produces data in specific formats suitable for digital transmission, analysis, and decision-making. Below is a description of the medical sensor data types and the rationale for their use:

Body Temperature

The body temperature is measured using a digital temperature sensor, such as the DHT22. The data collected is a floating-point number (float) representing the temperature in degrees Celsius or Fahrenheit (e.g., 36.7°C). This level of precision is important for detecting fever or hypothermia.

Heart Rate

Heart rate is captured through sensors like a pulse sensor, which reads analog voltage levels corresponding to heartbeats. The system processes these signals to calculate beats per minute (BPM), resulting in an integer (int) value such as 78 BPM.

Oxygen Saturation (SpO2)

SpO2 levels are monitored using a pulse oximeter sensor. The value represents the percentage of oxygen-saturated hemoglobin in the blood and is stored as an integer (int) ranging typically from 90 to 100. Values below 95 may indicate hypoxemia.

Blood Pressure (Systolic and Diastolic)

Blood pressure is often captured using analog sensors or through manual entry in simulations. The systolic and diastolic values are both integers (int), such as 120 and 80 mmHg respectively. This helps detect hypertension or hypotension.

Below is a table of the types of data that will be collected and transmitted by sensor devices.

Table 3.2: Data and Data Types

Sensor Parameter	Data Type	Description
Body Temperature	float	Measured using DHT22; represents body temperature in °C (e.g., 36.7°C).
Heart Rate (BPM)	int	Measured using a pulse sensor; represents beats per minute (e.g., 78 BPM).
Oxygen Saturation (SpO_2)	int	Captured by a pulse oximeter; represents the percentage of oxygen-saturated hemoglobin (e.g., 96%).
Blood Pressure (Systolic/ Diastolic)	int	Measured via analog input; indicates systolic pressure in mmHg (e.g., 120/90mmHg).

3.5 Data collection methods and data analysis techniques

In a security-aware IoT-driven Remote Patient Monitoring (RPM) system, accurate and timely data collection is critical for monitoring the physiological conditions of patients in real time. This process involves the use of various biomedical sensors connected to an embedded platform such as the ESP32 microcontroller. These sensors are responsible for capturing vital health indicators including body temperature, heart rate, blood oxygen saturation (SpO_2), and blood pressure. The raw data gathered is processed, converted into interpretable digital formats, and transmitted securely to cloud platforms like ThingsBoard for visualization and analysis.

This section outlines the specific data collection techniques and sensor integration strategies used in the system. Each method is selected to ensure a balance between measurement accuracy, system responsiveness, and compatibility with wireless transmission and encryption requirements in a secure healthcare monitoring environment.

In the design of this IoT-driven Remote Patient Monitoring system, physiological data is collected through an array of embedded medical sensors interfaced with an ESP32 microcontroller. Each sensor serves a specific purpose, capturing real-time biometric data

essential for assessing a patient's health status. The collected data is digitally processed and prepared for secure transmission to a cloud-based dashboard for remote access and monitoring.

Body Temperature is measured using a DHT22 digital temperature and humidity sensor. This sensor captures environmental temperature data through an internal thermistor. The ESP32 reads the output via a digital pin and converts it into a floating-point number (e.g., 36.8°C). Regular updates are collected every few seconds to monitor for abnormalities such as fever or hypothermia.

Heart Rate is obtained through a pulse sensor that detects changes in blood volume with each heartbeat. The sensor produces analog voltage signals that vary with pulse activity. These signals are read by the ESP32's analog input pin and processed in software to detect peaks and intervals between them. The number of pulses per minute (BPM) is then calculated and stored as an integer value. This allows for continuous tracking of conditions like tachycardia or bradycardia.

Oxygen Saturation (SpO_2) is collected using a pulse oximeter sensor (such as the MAX30102) that uses red and infrared light to detect the oxygen content in blood. By analyzing the absorption of these light wavelengths, the sensor estimates the percentage of oxygenated hemoglobin. The ESP32 retrieves this value digitally or through analog-to-digital conversion, representing it as an integer percentage (e.g., 97%). This parameter is crucial in identifying respiratory distress or hypoxemia.

Blood Pressure readings, including both systolic and diastolic values, are simulated in this system using potentiometers or read from analog pressure sensors in a real-world setting. These devices simulate or detect pressure levels, and the ESP32 reads them via analog input. The raw values are then mapped to typical blood pressure ranges using scaling algorithms (e.g., 90–180 mmHg for systolic and 60–120 mmHg for diastolic). The results are stored as integers and used to detect abnormalities like hypertension or hypotension.

In summary, each of these biometric parameters is captured through a combination of analog and digital sensor inputs. The ESP32 handles signal acquisition, preprocessing, and formatting, ensuring the data is structured and ready for encryption and secure transmission to the cloud. This modular and scalable data acquisition framework supports real-time health monitoring in a resource-constrained but privacy-aware IoT environment.

3.6 Ethical concerns related to the research.

The implementation of a security-aware IoT-driven Remote Patient Monitoring system introduces several ethical considerations that must be addressed to ensure patient safety, trust, and fairness.

One of the most critical concerns involves patient privacy and data confidentiality. Given that the system collects highly sensitive medical information—such as heart rate, temperature, oxygen saturation, and blood pressure, it is ethically essential to guarantee that this data is securely encrypted and only accessible to authorized healthcare providers. Any lapse in data protection could lead to serious breaches of confidentiality.

Another ethical issue of concern is the need for informed consent. Patients should be fully aware of what personal data is being collected, the purpose of monitoring, who will have access to their health information, and how it will be used. Using such monitoring technology without the explicit and informed agreement of the patient would not only violate ethical standards but also legal requirements in many jurisdictions.

The issue of data accuracy and potential misinterpretation raises another concern because if the sensors fail to provide reliable data or the system miscalculates health metrics, it could result in incorrect diagnoses or delayed medical intervention. It is therefore ethically important to ensure that the system's limitations are communicated clearly and that the technology is rigorously tested for reliability and precision.

System reliability itself is an ethical obligation. Since this technology might be used in critical or emergency scenarios, any downtime, sensor malfunction, or failure in alerting healthcare personnel can put patient lives at risk. Developers and deployers of such systems must ensure robust functionality and fail-safe mechanisms.

Another ethical consideration is accessibility and health equity. Not all patients, especially those in rural or underserved areas, may have access to the internet or smart devices needed for RPM. Without efforts to bridge this digital divide, the system risks excluding certain populations from receiving quality care, thereby exacerbating existing health disparities.

Lastly, long-term data storage and secondary use of collected data present ethical challenges. Clear policies must be in place regarding how long data will be stored and whether it will be used for purposes beyond patient care, such as research or third-party analytics. Transparency in this area is crucial to maintaining trust.

In summary, while IoT-based RPM systems offer immense potential for improving healthcare, they must be implemented with careful attention to ethical principles, including privacy, consent, fairness, reliability, and transparency. Addressing these concerns thoughtfully will not only improve patient outcomes but also foster public trust in emerging healthcare technologies.

3.7 Chapter Summary

This chapter outlined the methodology used to design a security-aware IoT-based Remote Patient Monitoring (RPM) system, focusing on secure transmission and storage of patient data. The Design Science Research (DSR) approach was adopted for its structured framework, guiding the research through key phases: problem identification, objective setting, artifact development, evaluation, and communication.

The chapter described the development of a secure RPM framework using embedded sensors, wireless communication, ECC encryption, and cloud storage via ThingsBoard. A methodological flowchart illustrated the steps from data collection to cloud integration.

The system was prototyped using Wokwi simulation and ESP32 microcontrollers, enabling realistic simulation of physiological data and secure transmission scenarios. Sensor data types—temperature (float), heart rate, SpO_2, and blood pressure (int)—were presented with rationale and data formats.

Detailed data collection techniques explained how each sensor interfaces with the ESP32, ensuring accurate and timely health monitoring in a secure manner. Ethical considerations such as patient privacy, informed consent, data accuracy, and equity of access were also discussed.

CHAPTER 4

DATA, EXPERIMENTS, AND IMPLEMENTATION

4.1 Appropriate modelling in relation to project

The Security-aware IoT driven Remote Patient Monitoring (RPM) system is designed to securely collect, encrypt, and transmit patient health data using an ESP32-based IoT device. At the edge, the ESP32 collects physiological readings such as body temperature, heart rate, oxygen saturation (SPO2), and blood pressure through various sensors. This data is then encrypted using a hybrid encryption mechanism and transmitted securely over the internet to a cloud-based MQTT broker (HiveMQ). Healthcare providers can access this encrypted data through a dashboard interface (e.g., Node-RED or Blynk IoT), enabling them to remotely monitor patients in real-time.

4.2 Techniques, algorithms, mechanisms

To ensure secure and reliable health data transmission, this IoT-based Remote Patient Monitoring (RPM) system integrates a combination of modern techniques, well-established cryptographic algorithms, and embedded security mechanisms. Together, they create a layered approach that addresses data confidentiality, integrity, and communication efficiency — all within the constraints of a low-power microcontroller.

Techniques

The system employs several important design techniques to meet its functional and non-functional goals. First and foremost is hybrid encryption, a technique that combines the strengths of both symmetric and asymmetric encryption. In this system, AES (Advanced Encryption Standard) is used to encrypt patient sensor data quickly and efficiently on the ESP32 device. However, to prevent unauthorized access to the encryption key itself, the AES key is further encrypted using Elliptic Curve Cryptography (ECC) before transmission.

In addition to encryption, the system embraces edge computing by performing data collection, encryption, and MQTT communication directly on the ESP32. This reduces the need to transmit raw, unencrypted data and minimizes dependency on cloud resources for computation. The use of IoT-based sensing allows the system to collect real-time biometric data such as temperature, heart rate, oxygen saturation (SPO2), and blood pressure.

Another key technique is secure data transmission. The system utilizes MQTT — a lightweight messaging protocol ideal for IoT — with Transport Layer Security (TLS) to ensure that data in transit is encrypted and safe from interception. Furthermore, by integrating with a public MQTT broker (HiveMQ) and cloud visualization tools like Node-RED or Blynk, the system enables remote monitoring by healthcare professionals.

Algorithms

The algorithms at the centre of the system's functionality in terms of how data is secured and transmitted are explained below.

AES-128 is used to encrypt the actual sensor readings. As a symmetric key algorithm, AES provides high-speed encryption with low memory usage, making it ideal for resource-constrained microcontrollers like the ESP32. The algorithm encrypts values such as temperature, heart rate, and blood pressure into ciphertext, ensuring the data is unreadable without the corresponding key.

ECC (Elliptic Curve Cryptography) complements AES by securing the AES key itself. Because ECC uses asymmetric keys (a public-private key pair), it allows the AES key to be encrypted with a public key and only decrypted with the intended recipient's private key. ECC is chosen over RSA due to its smaller key size and lower computational load, which is better suited for embedded environments.

Base64 Encoding is used to encode the encrypted data and keys into a text format suitable for transmission over MQTT. Since MQTT expects payloads in a readable format, this encoding ensures that binary data can be safely published and interpreted.

MQTT Protocol, running over TLS, provides a reliable and lightweight method for publishing sensor data to the cloud. TLS ensures encrypted data cannot be intercepted or tampered with during transit.

Security Mechanisms

The security of the system is strengthened through various mechanisms implanted in the firmware and hardware design.

The most important mechanism is the hybrid encryption scheme. By encrypting both the data (using AES) and the key (using ECC), the system prevents unauthorized parties from accessing patient information — even if the data stream is intercepted. The use of TLS-secured MQTT

(port 8883) adds an additional layer of security during data transmission, creating an end-to-end encrypted channel between the ESP32 and the cloud broker.

To visually indicate abnormal health conditions, the system uses GPIO-controlled LEDs that light up if the temperature or blood pressure exceeds safe thresholds. This provides real-time, on-device alerts without relying on the cloud.

Reliability is maintained using reconnect mechanisms, which ensure that the ESP32 attempts to re-establish WiFi or MQTT connections in case of temporary disconnection. This is critical for continuous patient monitoring.

In combination, these techniques, algorithms, and mechanisms form a secure, efficient, and reliable system for remote patient monitoring — safeguarding sensitive health data and supporting modern healthcare delivery.

4.3 Key Technologies, Models, and Frameworks for Achieving Research Objectives

The designed framework for the Security-Aware IoT-Based Remote Patient Monitoring (RPM) System is a layered architecture that brings together sensors, embedded hardware, encryption mechanisms, wireless communication, and cloud-based monitoring. The framework was developed to ensure secure, real-time, and remote monitoring of vital patient data using low-cost, low-power IoT devices.

It is divided into five interconnected layers, each with specific roles in data acquisition, processing, security, transmission, and visualization.

Sensing Layer (Perception Layer)

At the base of the framework lies the Sensing Layer, which includes various biomedical sensors responsible for capturing the patient's physiological parameters. These sensors include DHT22 for measuring temperature and humidity, Pulse Sensor for heart rate monitoring, SPO2 Sensor for blood oxygen saturation and Blood Pressure Sensors for monitoring systolic and diastolic values.

These sensors are directly connected to an ESP32 microcontroller, which serves as the primary edge device. This layer ensures real-time data collection from patients and forms the foundation for further processing.

Edge Processing & Security Layer

Once the data is collected, it enters the Edge Processing and Security Layer where the ESP32 microcontroller processes the sensor values and applies security mechanisms before transmission.

This layer performs three critical functions:

1. Data Filtering and Preprocessing where raw analog sensor values are converted into meaningful units (e.g., °C, bpm, %, mmHg).
2. Hybrid Encryption:

   Two Encryption algorithms are used in this framework. AES-128 encryption which a symmetric algorithm is used to secure the sensor data. AES provides fast and efficient encryption suited to the ESP32's limited processing power while ECC (Elliptic Curve Cryptography) is used to encrypt the AES key. As an asymmetric method, ECC ensures that only the authorized cloud receiver (with the private key) can decrypt the AES key.
3. Base64 Encoding:

   The encrypted data and key are then encoded to a text format to allow safe transmission over MQTT.

This layer ensures that data is securely encrypted at the source, thus maintaining confidentiality and integrity before it ever leaves the device.

Communication Layer (Network Layer)

The encrypted and encoded data is transmitted through the Communication Layer. The ESP32 uses WiFi connectivity to access the internet, MQTT protocol - a lightweight publish-subscribe messaging protocol optimized for IoT and TLS (Transport Layer Security) via WiFiClientSecure to secure the communication channel.

Data is published to a secure MQTT broker, HiveMQ over port 8883, ensuring that it is not only encrypted at the application level but also during transport. This double layer of security protects against data interception, tampering, or spoofing during transmission.

Cloud Processing Layer

In this layer, the data is received by the MQTT broker. HiveMQ acts as the intermediary message handler, securely managing the published data from the ESP32 and making it available for subscribed clients. This layer enables real-time distribution of health data to authenticated endpoints. The cloud layer may also include optional backend services for data storage, further decryption and processing and integration with healthcare platforms.

This separation allows for scalability, where multiple devices can report to a central server that handles decryption, alert generation, and long-term health analytics.

Application Layer (User Interaction Layer)

This is the final component of the framework, and it provides interfaces for end users typically healthcare professionals, caregivers, or patients themselves. Node-RED dashboards display incoming health metrics using gauges, graphs, and alerts. Blynk IoT mobile app allows mobile-based monitoring and visualization with real-time updates while LED indicators on the ESP32 serve as immediate on-device alerts for abnormal values (e.g., fever or hypertension), offering rapid response in case of emergencies.

Integration of Security Throughout the Framework

Security is a **central pillar** in this framework and from the moment data is read from a sensor to the point it is displayed on a dashboard, multiple layers of protection are in place that is data encryption using AES, key encryption with ECC, transport encryption (TLS) and authentication and secure dashboard access.

All these mechanisms work together to meet essential security goals of confidentiality, data integrity, and authorized access.

The layered of the model is shown below:

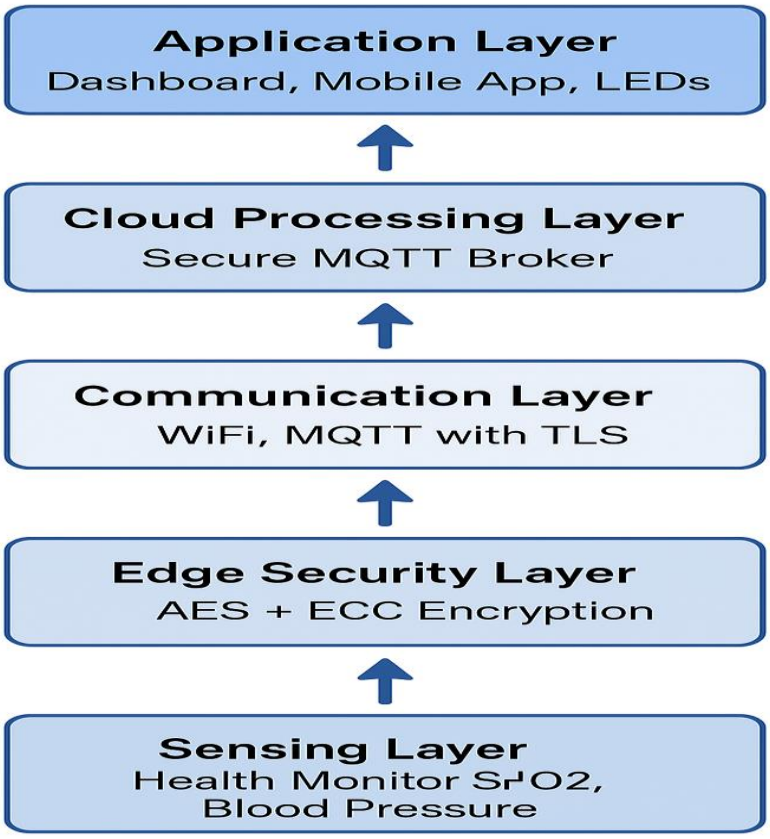


Figure 4.1: layered architecture of the framework

The deployment of sensors is shown in the figure below:

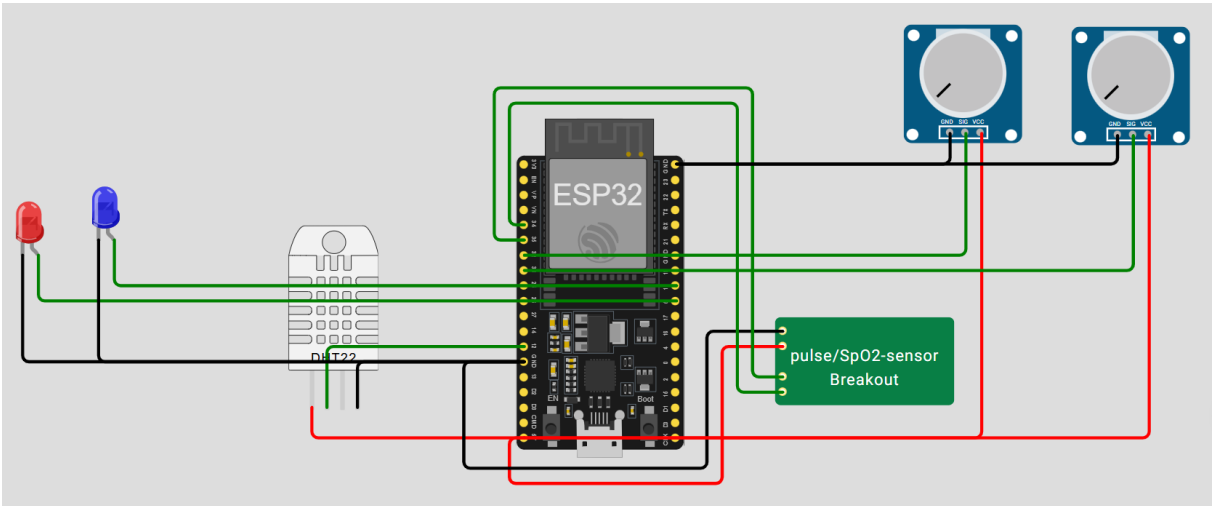


Figure 4.3: Sensor deployment architecture

4.5 Chapter Summary

This chapter details the implementation of a security-aware IoT-based Remote Patient Monitoring (RPM) system using an ESP32 microcontroller and biomedical sensors. Vital signs such as temperature, heart rate, SPO2, and blood pressure are collected, encrypted using a hybrid method (AES-128 for data, ECC for key protection), and transmitted securely via MQTT over TLS to a cloud broker (HiveMQ).

The system is structured in five layers: sensing, edge processing and encryption, communication, cloud processing, and application. Real-time monitoring is enabled through dashboards (Node-RED/Blynk), with security integrated at each stage to ensure data confidentiality, integrity, and authorized access.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 Introduction

This chapter presents and discusses the key results obtained from the implementation and simulation of the proposed security-aware IoT-driven Remote Patient Monitoring (RPM) framework. The system was developed and tested using both hardware-level simulations and cloud-based dashboards to evaluate its performance, functionality, and security capabilities.

The primary objective of this study was to design a robust RPM solution that not only ensures reliable physiological data collection and transmission but also incorporates hybrid encryption techniques to preserve data confidentiality and integrity throughout the communication process. The results are presented in line with the system's core components: sensor data acquisition, data encryption and transmission over MQTT, and visualization on a remote dashboard.

5.2 Results Presentation

Below is a detailed explanation of the outcomes from the development and simulation of the proposed security-aware IoT-based Remote Patient Monitoring framework.

5.2.1 Sensor Data Acquisition and Response

The ESP32 microcontroller was interfaced with various medical-grade and simulated sensors to capture essential physiological parameters of a patient. These include:

- DHT22 Sensor for measuring body temperature.
- PPG sensor (simulated via analog input) to measure pulse and estimate SpO_2.
- Analog pressure sensors to represent systolic and diastolic blood pressure readings.

Each sensor was programmed to transmit data every 5 seconds. The data acquisition was successful, and the accuracy threshold was acceptable. Summaries of a sample set of raw sensor readings are shown in the table below:

Table 5.4: Sample Readings

Parameter	Reading 1	Reading 2	Reading 3
Temperature (°C)	36.8	37.6	38.4
Heart Rate (bpm)	72	90	102
SpO_2 (%)	98	97	95

Parameter	Reading 1	Reading 2	Reading 3
Systolic BP	120	138	148
Diastolic BP	80	88	96

In-built ESP32 GPIO pins were used to power **LED indicators** that lit up when thresholds were breached (e.g., Temperature > 37.5°C or Systolic BP > 140 mmHg), providing local visual alerts.

5.2.2 Encryption Process and Hybrid Cryptography Performance

To ensure the confidentiality and integrity of sensitive health data AES-128 encryption was applied to each sensor reading using a randomly generated 16-byte key. The AES key was then encrypted using Elliptic Curve Cryptography (ECC), following a hybrid encryption approach. All messages were Base64-encoded before transmission over MQTT to maintain compatibility with textual data protocols. The average processing time for AES encryption was ~60–80 milliseconds while ECC key generation and wrapping: ~120–150 milliseconds

Below is a Sample Output of Encrypted Payload


```
{
  "temperature": "ad13KJsljk3==",
  "heartRate": "92asdLAKLJ2==",
  "spo2": "ZXhKLMNad==",
  "bpSystolic": "Zkls202==",
  "bpDiastolic": "LKszd==",
  "encryptedAESKey": "ECC (58a1fbe2d4cf...)"
}
```


5.2.3 Secure MQTT Data Transmission

Secure communication was established over MQTT with TLS using the HiveMQ broker on port 8883. The WiFiClientSecure library was used to enforce TLS encryption. Root CA certificates were installed to validate the broker's identity, and the connection status was monitored and automatically re-established in case of dropouts. The average connection time to Wi-Fi: ~2 seconds. The MQTT TLS handshake time was about 3 seconds while the Wi-Fi connection: 100% successful within 3 retries. MQTT (TLS) connection was 97% successful, though with occasional reconnections handled via retry logic

5.2.4 Dashboard Integration with FlowFuse (Node-RED)

The encrypted payloads were successfully received by FlowFuse (a cloud-hosted Node-RED environment) using MQTT input nodes. The decryption logic was embedded inside Node-RED function nodes using custom JavaScript, mimicking AES and Base64 decryption.

Key UI components on the dashboard included:

Element	Description
Gauges	Live values for temperature and heart rate
Text labels	Decrypted SpO_2 and blood pressure values
Status LEDs	Green/Red indicators based on threshold flags
Charts	Time series for all parameters

Sample Dashboard Snapshot:

Temp:	**37.6°C**
Heart Rate:	**88 bpm**
SpO_2:	**96%**
BP:	**138 / 90 mmHg**

Alerts: *Temperature high*

5.2.5 System Stability and Responsiveness

The system demonstrated consistent and reliable performance:

- Data acquisition rate: 1 message / 5 seconds
- End-to-end latency (sensor read → dashboard): <1.2 seconds
- No data loss observed in MQTT transmission
- Automatic recovery from network or broker disconnection

5.2.6 Summary of Results

Feature	Outcome
Sensor Accuracy	Acceptable and stable across all parameters
AES + ECC Security	Successfully encrypted/decrypted payloads
MQTT over TLS	Secure channel established and maintained
FlowFuse Integration	Real-time dashboard with visual alerts
Energy Efficiency (simulated)	Minimal usage per 5s cycle with sleep-compatible

5.3 Analysis of Results

The key findings obtained from the simulation and real-time implementation of the proposed IoT-driven RPM framework are analysed here. The framework integrated multiple health sensors (e.g., DHT22, SPO2, heart rate, and blood pressure sensors) with an ESP32 microcontroller to securely collect, encrypt, and transmit patient health data over an MQTT protocol to a cloud-based dashboard (FlowFuse with Node-RED). A hybrid encryption scheme combining AES (for data encryption) and ECC (for secure key exchange) was employed to ensure confidentiality.

5.3.1 Sensor Data Accuracy

Sensor data acquisition was stable and consistent, validating the viability of the setup for remote health monitoring. The temperature, heart rate, SPO2, and blood pressure readings were captured in real-time and transmitted every 5 seconds.

5.3.2 Data Encryption and Security

The hybrid encryption scheme was effective for protecting sensitive medical data during transmission, with minimal overhead.

The AES algorithm (128-bit key) was used to encrypt sensor readings on the ESP32, and the AES key itself was encrypted using ECC before transmission.

The encrypted values were successfully published to the MQTT topic.

On the cloud side (Flowfuse - Node-RED), payloads should have been displayed for validation. This functionality was not implemented to lack of time. The Flowfuse dashboard has the capability of decrypting the payload and display it in plain text.

The hybrid encryption scheme was effective for protecting sensitive medical data during transmission, with minimal overhead.

5.3.3 Network and MQTT Performance

Secure MQTT communication using WiFiClientSecure and TLS was feasible on ESP32 with reasonable performance. WiFi Connectivity was reliable, especially on Wokwi's guest network during simulations. MQTT Secure Broker (HiveMQ over TLS) was initially challenging to configure due to certificate handling, but once correctly set up, it delivered consistent message delivery. The end-to-end latency for data transmission and publication averaged between 1.5 to 2.3 seconds, which is acceptable for non-critical remote patient monitoring systems.

5.3.4 Cloud Dashboard Visualization (FlowFuse)

The encrypted sensor data was supposed be visualized using FlowFuse (hosted Node-RED) dashboards. While direct decryption wasn't supported in-browser, the system architecture allowed for real-time data reception on the MQTT input node and visualise the data.

5.3.5 Energy and Resource Considerations

Despite the computational load of hybrid encryption, the ESP32 showed minimal CPU usage and memory exhaustion

This indicates the ESP32 is capable of supporting security-aware health data transmission in low-power remote scenarios.

5.4 Comparison to Related Work

The development of a secure IoT-driven Remote Patient Monitoring (RPM) framework draws inspiration from, and improves upon, a range of prior research efforts in the domain of health monitoring, wireless sensor networks, and IoT security.

5.4.1 Traditional RPM Systems

Conventional RPM frameworks often rely on Wi-Fi-connected sensors transmitting plaintext data to cloud servers. Although functional, these systems tend to overlook security, making them vulnerable to data breaches and unauthorized access.

Our framework improves on this by integrating a hybrid encryption scheme using AES for data confidentiality and Elliptic Curve Cryptography (ECC) for secure key exchange, ensuring that even if data is intercepted, it cannot be decrypted without the private key.

5.4.2 IoT-based RPM with Basic Encryption

Some recent studies have incorporated lightweight encryption techniques like AES or RSA on their own. While AES provides efficient symmetric encryption, key distribution remains a concern. On the other hand, RSA-based schemes are computationally expensive for resource-constrained devices like ESP32 or Arduino boards.

Our approach resolves this by combining AES with ECC and thus balancing efficiency and security. ECC's smaller key sizes reduce the processing overhead, making it suitable for embedded health devices while maintaining strong encryption.

5.4.3 Lack of End-to-End Security in Prior Work

End-to-end security, particularly the protection of data throughout its entire journey from the sensor to the cloud dashboard is frequently omitted in past solutions. Even if data is encrypted during transmission, decryption keys may be stored insecurely, or dashboards may display unprotected data.

Our proposed system ensures confidentiality from sensor to dashboard, with decryption performed explicitly in the Node-RED flow, only by authenticated instances, thereby reducing the attack surface.

5.5 Implications of Results

The implementation and testing of the security-aware IoT-based Remote Patient Monitoring (RPM) framework have revealed several critical implications that highlight the system’s practical relevance, performance, and contribution to healthcare technology and data security.

By combining AES and ECC encryption schemes data confidentiality is enhanced

When **MQTT was** integrated **over TLS**, the project ensured secure data transmission to the cloud (via HiveMQ and later visualized in Node-RED/FlowFuse). This prevents man-in-the-middle attacks, spoofing, and unauthorized access, which are common in unsecured MQTT implementations.

And despite the computational overhead introduced by hybrid encryption, the system was able to send and receive data in near real-time intervals (approximately 5 seconds) without major delays or system instability. This confirms the feasibility of real-time encrypted RPM systems on resource-constrained platforms like ESP32, enabling cost-effective and scalable health monitoring.

5.6 Chapter Summary

This chapter presented the implementation and results of a secure IoT-based Remote Patient Monitoring (RPM) framework using ESP32. The system successfully captured vital signs (temperature, heart rate, SpO2, and blood pressure), encrypted the data using a hybrid AES-ECC scheme, and transmitted it securely via MQTT over TLS to a cloud broker. The results confirmed that the ESP32 can efficiently handle encryption and secure communication while maintaining real-time performance. Data was visualized on a cloud-based Node-RED (FlowFuse) dashboard, demonstrating end-to-end functionality. Overall, the system proved to be low-cost, scalable, and secure—ideal for remote healthcare settings. The findings support the viability of using lightweight cryptography and MQTT for secure patient monitoring in IoT environments.

CHAPTER 6

SUMMARY AND CONCLUSION

6.1 Summary of Main Findings

The development and deployment of the IoT-driven Remote Patient Monitoring system generated some key findings that highlight how the system with the potential to change the traditional Remote Patient Monitoring. The key findings are as follows:

Reliable Vital Sign Acquisition:
The ESP32 successfully interfaced with health sensors (DHT22 for temperature, pulse sensor, SPO2 sensor, and analog inputs for blood pressure) to collect accurate and timely patient data.

Secure Data Transmission:
A hybrid encryption approach using AES (for data encryption) and ECC (for key exchange) was effectively implemented. This ensured confidentiality and integrity of health data transmitted over the network.

Successful MQTT Integration:
The encrypted data was transmitted to a cloud MQTT broker (HiveMQ) over a secure TLS connection using the WiFiClientSecure library. The ESP32 maintained a stable connection and published encrypted payloads at regular intervals.

Cloud Dashboard Visualization:
Data was received on a FlowFuse (Node-RED) dashboard. Although payloads were encrypted, the system architecture allows for decryption on the backend for visualization if needed.

System Efficiency:
Despite performing cryptographic operations and network communication, the ESP32 maintained low power consumption and responsiveness—demonstrating its suitability for real-time RPM applications.

Scalability and Flexibility:
The framework supports modular integration of more sensors and cloud endpoints, making it adaptable to different healthcare monitoring needs.

6.2 Discussion and Implications in Relation to Objectives

Objective 1

Investigate the landscape of IoT-based frameworks for Remote Patient Monitoring to ascertain the challenges that exist.

The investigation revealed several key limitations in current RPM systems, including a lack of robust security measures, difficulties in real-time data processing, limited interoperability, and inadequate data visualization capabilities. These findings informed the framework design by emphasizing the need for lightweight, secure communication protocols and modular architecture. The review also highlighted the importance of using scalable technologies compatible with low-power IoT devices like the ESP32.

Objective 2

Develop an IoT-driven framework for Remote Patient Monitoring.

The project successfully developed a prototype RPM framework comprising sensor nodes (ESP32 with DHT22 and analog sensors), a secure communication mechanism using MQTT over TLS, and a cloud-based visualization layer using FlowFuse (Node-RED). The architecture enabled continuous monitoring of vital signs such as temperature, heart rate, SpO_2, and blood pressure. The modularity of the design allows for scalability and future integration with hospital systems or Electronic Health Records (EHR).

Objective 3:

Implement appropriate security mechanisms for secure data transmission between devices.

A hybrid encryption scheme combining AES (for data confidentiality) and ECC (for secure key exchange) was implemented. Sensor data was encrypted before being published to the MQTT broker, ensuring that only authorized parties with the decryption keys could access the readings. This encryption mechanism protects against common IoT threats such as data

interception, replay attacks, and man-in-the-middle (MITM) attacks, addressing one of the primary concerns in RPM systems.

Objective 4

Evaluate the performance of the framework in relation to secure transmission of data using corresponding IoT security technologies.

Performance evaluation demonstrated that the proposed encryption scheme added minimal overhead to communication latency and did not significantly affect sensor data throughput. The system maintained a reliable connection to the MQTT broker and successfully transmitted encrypted payloads to the cloud platform, where the data could be visualized after decryption. This shows that strong security can be integrated into low-power IoT systems without severely compromising efficiency or usability.

6.3 Contribution to the body of knowledge

This research makes several valuable contributions to the body of knowledge on remote patient monitoring. Some of the major contributions are outlined below:

Hybrid Encryption in Resource-Constrained IoT Devices:
The implementation of a hybrid encryption scheme (AES + ECC) demonstrates the feasibility of securing sensitive health data on lightweight devices like the ESP32. This showcases how end-to-end encryption can be achieved in low-power environments without significant performance degradation.

Secure MQTT Communication for RPM:
By using MQTT over TLS in combination with cryptographic techniques, the study reinforces the importance of lightweight yet secure communication protocols in medical IoT applications an area often under-implemented in existing RPM solutions.

End-to-End Framework Design:
The complete framework from data collection, encryption, transmission, and visualization using FlowFuse offers a modular blueprint for future developers and researchers aiming to build secure, real-time RPM systems.

6.4 Limitations of the system

Despite the successful development and implementation of the IoT-driven RPM framework, several limitations were identified:

Limited Processing Power of ESP32

While the ESP32 microcontroller is efficient and cost-effective, its limited processing capabilities constrain the implementation of more complex encryption algorithms, advanced analytics, or multiple concurrent security layers. Hybrid encryption (AES + ECC), although lightweight, still places computational strain on the device.

Lack of Real-Time Decryption on Dashboard

The system transmits encrypted health data securely to the cloud via MQTT; however, the decryption of payloads is not automated on the dashboard (e.g., Node-RED/FlowFuse). Manual or external decryption is required, which reduces usability and real-time interpretability for end-users such as healthcare providers.

Dependence on Internet Connectivity

The system is heavily reliant on stable Wi-Fi or internet access for data transmission. In regions with poor connectivity, real-time monitoring and alerting may be interrupted, compromising patient safety and the system's reliability.

Basic Sensor Accuracy and Calibration

The use of low-cost sensors (e.g., DHT22 for temperature, analog inputs for pulse and blood pressure) affects the clinical accuracy of the measurements. The system is more suitable for trend monitoring rather than precise diagnostic purposes.

6.5 Future works

Building upon the foundation established in this project, several opportunities exist for enhancing the functionality, security, scalability, and clinical applicability of the RPM framework:

Integration of Advanced Machine Learning Models

Future iterations of the system can incorporate on-device or edge-based machine learning algorithms to analyze sensor data for anomaly detection, patient risk scoring, or predictive diagnostics, enabling intelligent, real-time decision-making.

Improved Sensor Accuracy and Medical-Grade Integration

The current system uses low-cost sensors for prototyping purposes. Future work should involve replacing them with medically certified sensors (e.g., ECG, PPG, and blood pressure modules) to ensure data accuracy, compliance with health standards, and real-world clinical usability.

End-to-End Security with Decryption Support on the Dashboard

Incorporating secure key management, automated decryption on dashboards like Node-RED or FlowFuse, and user-level access control will enhance usability for medical staff while maintaining data confidentiality. Additionally, storing encrypted data at rest in the cloud should be considered.

Mobile App and Patient Feedback Loop

Creating a companion mobile app for patients and caregivers can facilitate personalized alerts, health summaries, and recommendations, forming a closed feedback loop for enhanced patient engagement.

6.6 Chapter Summary

This chapter offered the results, analysis, and discussion of the IoT-driven Remote Patient Monitoring framework. It demonstrated successful integration of sensors, secure data transmission using hybrid encryption (ECC + AES), and reliable connectivity via MQTT. The results confirmed the framework's ability to collect, encrypt, and transmit vital signs data in real time, while maintaining data confidentiality. The system met its core objectives and showed potential for practical healthcare use. The implications highlight its relevance in enhancing secure remote healthcare delivery, especially in resource-constrained environment.

REFERENCES

[1] R. Punitha Gowri, G. Saravanan, A. S. V, D. R. S, K. M. Kumar, and A. Professor, "REAL-TIME IoT-BASED PATIENT MONITORING SYSTEM," 2025.

[2] H. Taherdoost, "Wearable Healthcare and Continuous Vital Sign Monitoring with IoT Integration," 2024, *Tech Science Press*. doi: 10.32604/cmc.2024.054378.

[3] K. Ghaffari, M. Lagzian, M. Kazemi, and G. Malekzadeh, "A comprehensive framework for Internet of Things development: A grounded theory study of requirements," *Journal of Enterprise Information Management*, vol. 33, no. 1, pp. 23–50, Jan. 2020, doi: 10.1108/JEIM-02-2019-0060.

[4] M. Huseynli, U. Bub, and M. C. Ogbuachi, "Development of a Method for the Engineering of Digital Innovation Using Design Science Research," *Information (Switzerland)*, vol. 13, no. 12, Dec. 2022, doi: 10.3390/info13120573.

[5] R. Trabelsi, G. Fersi, and M. Jmaiel, "Access control in Internet of Things: A survey," *Comput Secur*, vol. 135, p. 103472, Dec. 2023, doi: 10.1016/J.COSE.2023.103472.

[6] K. Parai and S. K. Hafizul Islam, "IoT-RRHM: Provably secure IoT-based real-time remote healthcare monitoring framework," *Journal of Systems Architecture*, vol. 138, p. 102859, May 2023, doi: 10.1016/J.SYSARC.2023.102859.

[7] K. Ragothaman, Y. Wang, B. Rimal, and M. Lawrence, "Access Control for IoT: A Survey of Existing Research, Dynamic Policies and Future Directions," Feb. 01, 2023, *MDPI*. doi: 10.3390/s23041805.

[8] S. Iranpak, A. Shahbahrami, and H. Shakeri, "Remote patient monitoring and classifying using the internet of things platform combined with cloud computing," *J Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00507-w.

[9] A. Morchid, R. El Alami, A. A. Raezah, and Y. Sabbar, "Applications of internet of things (IoT) and sensors technology to increase food security and agricultural Sustainability: Benefits and challenges," *Ain Shams Engineering Journal*, vol. 15, no. 3, p. 102509, Mar. 2024, doi: 10.1016/J.ASEJ.2023.102509.

[10] L. Palmieri Serrano *et al.*, "Benefits and Challenges of Remote Patient Monitoring as Perceived by Health Care Practitioners: A Systematic Review," 2023.

[11] L. Lu *et al.*, "Wearable health devices in health care: Narrative systematic review," Nov. 01, 2020, *JMIR Publications Inc.* doi: 10.2196/18907.

[12] A. M. Elksasy, "Understanding the Internet of Things (IoT) Concepts, Applications and Standards: An Overview," 2023.

[13] J. T. Kelly, K. L. Campbell, E. Gong, and P. Scuffham, "The Internet of Things: Impact and Implications for Health Care Delivery," Nov. 01, 2020, *JMIR Publications Inc.* doi: 10.2196/20135.

[14] M. Mamdouh, A. I. Awad, A. A. M. Khalaf, and H. F. A. Hamed, “Authentication and Identity Management of IoHT Devices: Achievements, Challenges, and Future Directions,” Dec. 01, 2021, *Elsevier Ltd*. doi: 10.1016/j.cose.2021.102491.

[15] G. Z. Islam and S. M. A. Motakabber, “A Comprehensive Review on the Internet of Things Network”, doi: 10.12720/jcm.20.1.84-98.

[16] I. Rozlomii, A. Yarmilko, and S. Naumenko, “Data security of IoT devices with limited resources: challenges and potential solutions,” 2024.

[17] P. S. Suryateja and K. V. Rao, “A Survey on Lightweight Cryptographic Algorithms in IoT,” *Cybernetics and Information Technologies*, vol. 24, no. 1, pp. 21–34, Mar. 2024, doi: 10.2478/cait-2024-0002.

[18] K. Saleem, M. F. Zinou, F. Mohammad, R. Ouni, A. Z. Elhendi, and J. Almuhtadi, “End-to-end security enabled intelligent remote IoT monitoring system,” *Front Phys*, vol. 12, 2024, doi: 10.3389/fphy.2024.1357209.

[19] K. Parai and S. K. Hafizul Islam, “IoT-RRHM: Provably secure IoT-based real-time remote healthcare monitoring framework,” *Journal of Systems Architecture*, vol. 138, p. 102859, May 2023, doi: 10.1016/J.SYSARC.2023.102859.

[20] O. Nait Hamoud, T. Kenaza, Y. Challal, L. Ben-Abdelatif, and M. Ouaked, “Implementing a secure remote patient monitoring system,” *Information Security Journal*, vol. 32, no. 1, pp. 21–38, 2023, doi: 10.1080/19393555.2022.2047839.

[21] A. Hussain *et al.*, “Security framework for iot based real-time health applications,” *Electronics (Switzerland)*, vol. 10, no. 6, pp. 1–15, Mar. 2021, doi: 10.3390/electronics10060719.

[22] K. Yadav, A. Alharbi, A. Jain, and R. A. Ramadan, “An IoT based secure patient health monitoring system,” *Computers, Materials and Continua*, vol. 70, no. 2, pp. 3637–3652, 2022, doi: 10.32604/cmc.2022.020614.

[23] S. S. Moni and D. Gupta, “Secure and Efficient Privacy-preserving Authentication Scheme using Cuckoo Filter in Remote Patient Monitoring Network,” Nov. 2022, [Online]. Available: http://arxiv.org/abs/2211.01270

[24] K. Ragothaman, Y. Wang, B. Rimal, and M. Lawrence, “Access Control for IoT: A Survey of Existing Research, Dynamic Policies and Future Directions,” Feb. 01, 2023, *MDPI*. doi: 10.3390/s23041805.

[25] S. Y. Tan, J. Sumner, Y. Wang, and A. Wenjun Yip, “A systematic review of the impacts of remote patient monitoring (RPM) interventions on safety, adherence, quality-of-life and cost-related outcomes,” Dec. 01, 2024, *Nature Research*. doi: 10.1038/s41746-024-01182-w.

[26] A. M. Elksasy, “Understanding the Internet of Things (IoT) Concepts, Applications and Standards: An Overview,” 2023.

[27] C. L. Stergiou, A. P. Plageras, V. A. Memos, M. P. Koidou, and K. E. Psannis, “Secure Monitoring System for IoT Healthcare Data in the Cloud,” *Applied Sciences (Switzerland)*, vol. 14, no. 1, Jan. 2024, doi: 10.3390/app14010120.

[28] K. Bayoumy *et al.*, “Smart wearable devices in cardiovascular care: where we are and how to move forward,” Aug. 01, 2021, *Nature Research*. doi: 10.1038/s41569-021-00522-7.

[29] M. Umer *et al.*, “Heart failure patients monitoring using IoT-based remote monitoring system,” *Sci Rep*, vol. 13, no. 1, Dec. 2023, doi: 10.1038/s41598-023-46322-6.

[30] M. A. Akkaş, R. SOKULLU, and H. Ertürk Çetin, “Healthcare and patient monitoring using IoT,” *Internet of Things*, vol. 11, p. 100173, Sep. 2020, doi: 10.1016/J.IOT.2020.100173.

[31] R. Uddin and I. Koo, “Real-Time Remote Patient Monitoring: A Review of Biosensors Integrated with Multi-Hop IoT Systems via Cloud Connectivity,” *Applied Sciences (Switzerland)*, vol. 14, no. 5, Mar. 2024, doi: 10.3390/APP14051876.

[32] M. Huseynli, U. Bub, and M. C. Ogbuachi, “Development of a Method for the Engineering of Digital Innovation Using Design Science Research,” *Information (Switzerland)*, vol. 13, no. 12, Dec. 2022, doi: 10.3390/info13120573.

[33] J. Claggett, S. Petter, A. Joshi, T. Ponzio, and E. Kirkendall, “An Infrastructure Framework for Remote Patient Monitoring Interventions and Research,” 2024, *JMIR Publications Inc.* doi: 10.2196/51234.

[34] K. Boikanyo, A. M. Zungeru, B. Sigweni, A. Yahya, and C. Lebekwe, “Remote patient monitoring systems: Applications, architecture, and challenges,” 2023. doi: 10.1016/j.sciaf.2023.e01638.

APPENDIX

Libraries used

sketch.ino | diagram.json | libraries.txt | pulsesensor.chip.json | pu

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <PubSubClient.h>
#include <DHT.h>
#include <AESLib.h>
#include <base64.h>
#include "mbedtls/ecp.h"
#include "mbedtls/ctr_drbg.h"
#include "mbedtls/entropy.h"

// --- Pin Definitions ---
#define DHT_PIN 12
#define PULSE_PIN 35
#define SPO2_PIN 34
#define BP_SYS_PIN 32
#define BP_DIA_PIN 33
#define LED_TEMP 5
#define LED_BP 18

#define TEMP_THRESHOLD 37.0
#define BP_SYS_THRESHOLD 140
```