MAAMSIC: MULTIMODAL AUTHENTICATION AND AUTHORIZATION MODEL FOR SECURITY OF IoT COMMUNICATION VIA GSM MESSAGING IN SUB-SAHARAN AFRICA

JOAN NABUSOBA

MASTER OF SCIENCE

(Information Technology)

JOMO KENYATTA UNIVERSITY

OF

AGRICULTURE AND TECHNOLOGY

2024

MAAMSIC: Multimodal Authentication and Authorization Model for Security of IoT Communication via GSM Messaging in Sub-Saharan Africa

Joan Nabusoba

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Information Technology of the Jomo Kenyatta University of Agriculture and Technology

2024

DECLARATION

This thesis is my original work and has not been presented for a degree in any other University

Signature: ……. Date: ……

Joan Nabusoba

This thesis been submitted for examination with our approval as the University Supervisors

Signature: ……. Date: ……

Professor Wilson Cheruiyot. PhD

JKUAT, Kenya

Signature: ……. Date: ……

Dr. Dennis Kaburu, PhD

JKUAT, Kenya

DEDICATION

This research thesis is dedicated to Almighty God, my parents, my brothers and sisters for their support. I also dedicate it to my business partner and mentor, for encouragement and support.

ACKNOWLEDGMENT

I would like to express my gratitude to my supervisors Dr. Dennis Kaburu and Professor Wilson Cheruiyot for their professional and scholarly guidance, critical review, availability and input which has jointly enabled me write this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

ACCRONYMS AND ABBREVIATIONS

MAAMSIC	Multimodal Authentication and Authorization Model for the Security of IoT Communication via GSM Messaging
Auth	Authentication
App	Application
JSON	JavaScript Object Notation
JWT	JSON Web Tokens
JWS	JSON Web Signatures
HTTP	Hyper Text Transfer Protocol
API	Application Programming Interface
CSRF	Cross-Site Request Forgery
XSS	Cross-Site Scripting

ABSTRACT

Internet of Things (IoT) which consists of heterogeneous devices is an enabling technology that can greatly improve the quality of lives in Sub-Saharan Africa. For instance, soil humidity and irrigation for e-agriculture, energy consumption, or even health data. As with technology, however, IoT has introduced security and privacy challenges. IoT devices create, transfer, process, and store sensitive data that must be protected from unauthorized access. Similarly, the devices and infrastructure linking with IoT and the IoT devices themselves are assets that must be protected. Though IoT devices are being adopted, there isn't wide access to GPRS In rural parts of Africa; hence users need to have access to technology that is seamless, viable, and easy to use. This thesis introduces a Multimodal Authentication and Authorization Model for the Security of IoT Communication via GSM Messaging (MAAMSIC). The MAAMSIC introduces a secure GSM multimodal authentication and authorization model that utilizes modalities of OAuth, JWT, JWS and 2-factor verification, and encrypts data and tokens between the user, the server and the IoT device. The model is illustrated using experiment methodology, which consists of a soil and moisture control system, phone app to send Attention Commands to IoT, and the server system. The system has a server to securely store encrypted data and send feedback to the user in case of any anomalies, as a way of fast mitigation. For validation, the system is compared with existing IoT systems that use GSM, and parameters such as request time, security and availability were tested. It was found out that MAAMSIC greatly improved the security of IoT communication with its added layer of security, which is lightweight and can be used in areas with low network coverage. The system provides the availability of both Internet phones and simple phones that don't use the internet. To ensure that third-party apps don't read data from the smartphone, it has an app to send commands, and the data JWT data is encrypted using custom JSON Web Encryption algorithm at the header. To advance MAAMSIC, there will be need for dedicated servers, integration of AI and data analytics to make it more efficient for integrating it with predictive models.

CHAPTER ONE

INTRODUCTION

1.1 Research Background

Internet of things (IoT) is a new paradigm that relies on widely spread connected objects that cooperate to automate many everyday actions (Rak, Salzillo & Romeo, 2020). Innovations in IoT are improving lives in rural Africa. For example, to solve water problems, Ingram & Memon (2020) are utilizing IoT technology to scale water supply systems in sub-Saharan Africa for sustainable water supply.

The Internet of Things (IoT) has been applied in numerous spaces, like smart home, drones, health, smart city, smart farming, among different areas. The IoT permits physical objects communicate with each other, share data and coordinate decisions. The IoT changes traditional objects into intelligent objects through exploitation of its enabling technologies Internet protocols, communication technologies, and sensor networks (Ferrag et al, 2020).

According to Mohamed *et* al (2020), in order to build up a green IoT-based horticulture solution, there are six fundamental challenges, including, equipment, data and information analytics, maintenance, versatility, infrastructure, information security, and privacy. The challenges in hardware concern the selection of sensors and meters for IoT gadgets. Thusly, there are different sorts of sensors types that can be utilized in IoT application (e.g., proximity sensor, temperature sensor, pressure sensor, chemical sensor, water quality sensor, gas sensor, moisture sensor… and so forth) The challenge in data analytics concern the utilization of prescient calculations and AI (like, deep learning approaches) in IoT information to get a nutritive answer for smart agribusiness. The challenge in maintenance concerns sensors checks of all devices in IoT since they can be easily harmed in the agriculture field. The challenge in mobility concerns the wireless communication type (e.g., 2G, 3G, 4G, 5G, WiFi), that are responsible for connection of sensors dispersed

over a huge area in the agricultural field. The challenges in infrastructure concern the installation and maintenance of the network infrastructure of IoT using new and developing technologies like network visualization, cloud computing, fog computing among others. With the adaption of green IoT-based agribusiness, an unexpected vulnerability may enter into the system, like injection of false data, raising new privacy and security issues hence leading to requests for a more secure communication system in the smart agribusiness field (Mohamed *et al*, 2020).

The knowledge and deployment of IoT systems in agriculture shall benefit Africa towards achieving zero hunger, according to Iorliam, Iorliam & Bum (2020). The authors recommend that government institutions, agricultural entrepreneurs, farmers and agricultural research institutes should leverage on the benefits of IoT systems in agriculture to increase crop production, monitor farms in real time remotely, monitor agricultural equipment which improves cost of operation and improve livestock and crop farming. Application of IoT in agriculture is a technology that is already utilized in America, Asia and Europe to solve their agricultural challenges. If African nations adopt these techniques with their large tracks of land, water supply from rivers and rain reservoirs, poultry and livestock, there will be high productivity hence achieving zero hunger target as well as optimize energy usage in farms (Iorliam, Iorliam & Bum, 2020).

To take care of a developing worldwide population projected to increment to 9 billion by 2050, production of food should increase from its present level. The majority of this growth should come from smallholder farmers who depend on generational knowledge in their cultivating practices and who live in areas where seasons and weather patterns are getting less predictable because of environmental change. The expansion of IoT is increasing the freedom to apply digital devices and services on smallholder farms, which includes soil and plants monitoring for agriculture, water quality in hydroponics, and ambient conditions in nurseries. In blend with other food security efforts, IoT-enabled precision for smallholder farming can possibly improve livelihood and speed up low and middle-income nations' journey to self-reliance (Antony *et al,* 2020).

According to Fastellini & Schillaci (2020), smart farming in Africa was impeded by advancement of ICT systems around the continent. Recent projects, like the Last Mile Project and other projects that interface sub-Saharan Africa to the world Internet have sparked researchers and different partners in agribusiness to take an action on digitization of agriculture and smart farming. These endeavors will go so high when Internet costs goes down and gets accessible to the greater most farmers. Farmers in Africa are ready to receive and adopt innovation that would improve on their day-by-day economic activities. That is the reason why mobile money in phones is so well known in Africa and helpful among farmers.

According to BRCK, about 800 million Africans do not have internet access making internet coverage, especially in rural areas, low (brck.com). As much as the acquisition of phones is growing in Africa, smartphone adoption is modest. Common types of mobile devices owned are basic phones. Worldwide, sub-Saharan Africans report the lowest ownership rate of smartphones compared to any other geographic region. Additionally, according to M-Kopa (https://m-kopa.com/impact/), one of the leading solar home system states that 75% of sub-Saharan Africa remains unconnected to the internet. Sending IoT data can use general packet radio service (GPRS), which is an advancement of GSM. However, since GPRS is not dedicated for transmission of IoT data due to weaknesses in power efficiency and coverage, GSM is preferred (Bima, Suryani & Wardana, 2020).

Agriculture is leading as the only field industry where the recent technology improvements haven't been adopted in large numbers. One of the main reasons that led to this issue is poor inert condition of farmers who reside in developing countries. The shortage of agricultural products is growing daily due to urbanization and overpopulation. The case of overpopulation makes the demand of agricultural resources to rise which demands better growth of farming products but due to globalization, big industries take huge portions of agricultural land and utilize it with non-agricultural activities. Basically, rural farming area is decreasing leading to decrease in agricultural development and resources. Therefore, there is an urgent need to critically look at the situation and giving emphasis to improve crop yields by utilizing the resources moderately without wasting

existing resources. Smart farming and use of IoT is one way of looking at it (Gupta *et al*, 2020).

There is explosive development of IoT, spurring a massive demand for many smart devices to access the wireless networks concurrently (Priyanka et al., 2020). It is predicted that over 20 billion devices will connect to wireless internet by the end of 2023 (Xu, Hu & Li, 2020).

Recent technological advancements in area pertinent to IoT works with a simpler adoption and utilization of smart farming with IoT. Such technological improvements incorporate, for instance, network communications, decrease of equipment size, power consumption optimization and access to cheap devices. Besides, the World's biggest agricultural producers are advancing the use of IoT in smart farming by making incentive projects and public policies to finance training and research (Navarro, Costa & Pereira, 2020).

Owuor, Laurent & Orero (2020) state that there is rapid increase in IoT applications in almost every area of our society. However, according to Franklin (2020), these increases of IoT startups fail to profit from their IoT innovations. This is due to security related problems like authentication and attack on privacy during IoT development, which are often underestimated or overlooked (Selgert, 2020). Even with inadequate coverage of the internet in rural parts of Africa, smart farming innovations are still being implemented using IoT and wireless devices (Olivera-Jr et al., 2020).

The use of technology still presents security issues to the African continent. Despite network coverage challenges, Ndubueze (2020) suggested that the number of cybercrime and victimization incidents is on the rise. However, the African continent has been slow in responding to crime and disorder in cyberspace and is still establishing cybercrime legislations. Hence, this makes Africans vulnerable in cyberspace, without the assailants facing the consequences, since there isn’t proper establishment and enforcement of cyber legislations (Ndubueze, 2020).

There has been several usages of multimodal authentication and authorization to make IoT efficient. Ammour & Alajlan, (2023) introduces a multimodal approach of using Electrogram Signal and fingerprints to incorporate biological traits, with the aim of preventing spoofing attacks. There has also been the use of AI to enable authentication in IoT. Shelke et al, (2022), presented a model that uses artificial intelligence-based algorithms, fingerprint recognition, RFID authentication and manual keypad entry in their authentication system. All these modules are interconnected to take global decisions about a person's authenticity in their systems (Shelke et al, 2022). Additionally, there has been a development of an AI-based multimodal biometric authentication model for single and group-based users' device-level authentication that increases protection against the traditional single modal approach (Farhad et al, 2023). The authors provided a comprehensive overview of all three authentication techniques with all the performance metrics (Accuracy, Equal Error Rate (EER), False Acceptance Rate (FAR)), security, privacy, memory requirements, and usability (Acceptability by user)) that will help one choose a perfect authentication technique for an application. In addition, the study also explores the performance of multimodal and multi-factor authentication and the application areas of authentication (Singla & Verma, 2023).

1.2 Statement of the Problem

IoT acquisition is Africa is growing rapidly with the aim of utilizing it in smart farming, to manage resources and improve yields with the aim of increasing food production in the ever-growing population. However, during development and acquisition of IoT systems, security-related problems are often overlooked or underestimated, making IoT startups fail to profit from these IoT innovations. Second, half of sub-Saharan Africa don't have access to electricity, majorly in rural areas. This hinders GPRS as a choice of IoT infrastructure, affecting its adoption.

In this thesis, we reviewed the current IoT systems in Africa that uses GSM due to lower coverage, and how they have handled the security aspect during data transmission. We utilized the use of GSM 2G mobile phones with no internet, and smartphones with

internet, to have an option for both rural farmers with no coverage of internet and power, and those with smartphones. We recorded the authentication results, and server request and response parameters. Our objective was to determine the algorithms used by the current GSM IoT systems are protecting their data, and create a model to improve the security problems and analyze its performance.

In this thesis, we developed experiment model with security implementation in authentication and authorization of modalities of OAuth, JSON Web Tokens, JSON Web Signature and JSON Web Encryption; and compared these to current GSM IoT systems for validation

1.3 Justification of the Study

To address and clarify our comprehension of Authentication and Authorization as a way of securing IoT systems, this research developed a Multimodal Authentication and Authorization Model for the Security of IoT Communication via GSM Messaging (MAAMSIC). The proposed model addresses the issues of security flaws that might hinder the acquisition of IoT systems in Africa, which is a great technology for increase of food production. Secondly, due to poor network coverage in major parts of sub-Saharan Africa making GPRS devices hard to use, MAAMSIC will utilize both simple phones and smart phones to send AT commands to a secured server via SMS, which will be relayed to the IoT. Additionally, smartphone devices will be used by having a smartphone app secured with JWT and JWS, which will send commands to the secure server. MAAMSIC gives access to users who either have a smartphone or a simple phone, a secure server for transmission of AT commands and an adoptable and reliable user interface. A combination of these technique is convenient, reliable and user-focused while ensuring security in data transmission between devices, the server and the IoT

1.4 Objectives of Study

1.4.1 Main Objective

The objective of this thesis was to construct a multimodal technique for securely sending and receiving data from IoT systems using GSM in sub-Saharan Africa (MAAMSIC).

1.4.2 Specific Objectives

The specific objectives of this thesis include:

1. To identify vulnerabilities previous GSM IoT systems in sub-Saharan Africa in the way they tackled the security, availability and reliability aspect.
2. To analyze authentication functionalities provided in the previous studies of GSM IoT Systems
3. To design and develop an experiment based Multimodal Authentication and Authorization Model for the Security of IoT Communication via GSM Messaging (MAAMSIC) system to enhance IoT data security
4. To develop and test and validate the MAAMSIC model with the aim of comparing it with existing IoT systems in previous research

1.5 Research Questions

1. What are the security issues facing IoT-based models in sub-Saharan Africa?
2. How can the existing models be used to inspire the new and more efficient MAAMSIC model?
3. How will constructing the new model guarantee authentication and authorization security?
4. What are the functionalities provided by the MAAMSIC model that will enhance security?

1.6 Scope of the Study

This research analyzed previously used IoT systems in sub-Saharan Africa and then constructed a new Multimodal Authentication and Authorization Model for the Security of IoT Communication via GSM Messaging (MAAMSIC). The study was based on a secure server that processes IoT commands and user commands, SMS Gateway, a reliable simulated IoT device (Iotify) chipped with GSM, soil moisture sensor and water level sensor, and mobile phones. To ensure that we exhausted both scenarios during validation, both simple phones that don't have access to GPRS and smart phones were used.

Data was collected from GitHub, and scholarly articles. GitHub provides a wide range of software libraries provided by major tech giants like Google, and a community to discuss issues whenever faced with one. Data collected led to the construction of a MAAMSIC model that was created, tested and validated. The research didn't dive into software maintenance due to limited budget, but the model was clear for ease of adaptability, scaling and reconstruction.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter will examine literature related to IoT and security. This will include definition of concepts, services, related work and gaps, knowledge contribution and later summary of the chapter. It also helps to better comprehend the questions related to the utilization of IoT in agribusiness in the 21st century.

2.1.1 Smart Farming

This can be characterized as the utilization of beneficial advancements to agricultural production strategies to help limit waste and lift efficiency. For such, smart farms utilize technological resources that help in different phases of the process of production, like plantation monitoring, management of soil, water system, pest control, tracking delivery, and so forth (Bhagat et al, 2019). Such resources incorporate, among others, temperature, humidity, pressure, ground compound concentration, automated flying gear, camcorders, farming information management systems, global positioning system (GPS) and networks (Stočes et al, 2016).

The combination of innovative resources into the farming production process is an applicable issue. From a financial perspective, the precision farming business sector is expected to have an income of US$10 billion of every 2023 with promising circumstances for technology suppliers, agricultural equipment and machinery suppliers, producers and others engaged with this business. Moreover, smart farms are relied upon to have the option to enhance food production by improving the utilization of supplements to the soil, decreasing the measure of pesticides and water consumption in water system (Kite-Powell, 2016).

2.1.2 Internet of Things (IoT)

IoT can be perceived as a network of interconnected intelligent devices fit for communicating with one another, creating significant information about the environment in which they work. Consequently, virtually any device equipped for building up a connection with the Internet can be viewed as a "thing" inside the setting of IoT, like domestic devices, hardware, furniture, agricultural or industrial machinery and even individuals (Madakam, 2015).

Although the idea of IoT isn't new, its adoption has expanded as of late, essentially on account of the advancement of technologies that support it, among which the improvement of equipment—with the resulting decrease in size and consumption of power—enhancements in connectivity with the Internet and between devices through wireless connection, big data, artificial intelligence and cloud computing. Every one of these technological components help construct a network of devices for sharing information and data, and acting effectively based on network inputs (Verdouw et al, 2019).

According to Vendouw et al (2019), the engineering of IoT systems is like the design of other computer frameworks. However, it should consider the particularities of this paradigm, for example, the restricted computing abilities of the devices, identification, detection and control of remote objects.

2.1.3 Intelligent Agriculture

The IoT design proposed by Chen, J & Yang (2019) and appears in Table 1, presents four layers, putting in the consideration of the principal components of an IoT solution: application, network, devices and services.

Table 2.1: IoT Architecture that Includes 4 Layers: Application, Processing, Transport and Perception Based on Navarro

Application	Monitoring farms, pest and disease control, irrigation etc.
Processing	Storage of Data, data filtering, analysis of data, data processing among others
Transport	Network protocols, application protocols
Perception	Sensor nodes, GPS

Source: (Costa & Pereira, 2020)

The perception layer identifies with the actual devices in the solution and how they interface with one another and with the transport layer. These devices are responsible for gathering information, enabling the communication of the supposed "things". This should be possible by utilizing business solutions, for example, UAV devices (Uddin et al, 2018), sensor hubs (Liu et al, 2016)—or new devices, created with segments like sensors and single-board computers (SBC)— like Arduino or Raspberry Pi—to assemble sensor hubs and communication gateways. Sensor hubs, for instance, are utilized to screen plant infections (Thorat et al, 2017), control ecological factors in greenhouses (Rivas-Sánchez et al, 2019) and external crops (Navurul, S & Prasad, 2017), among others. The connection between the devices that have a place with the perception layer and the services that belong to the processing layer is intermediated by the transport layer and may happen severally, for example, through the direct communication between sensor hubs and a data processing platform, (for example, FIWARE) or through a gateway that, other than intermediating the communication between sensor hubs and the web, goes about as an information center and enables the communication between network protocols that are initially incompatible, for example the Internet (Karim et al, 2017).

The transport layer alludes to the network and transport capabilities, for example, network and application protocols (Chen, J & Yang, 2019). IoT solutions utilize network protocols to enable communication between the processing layer and the perception layer. These protocols are utilized to make the wireless sensor networks (WSN), which permits wireless communication between sensor hubs and applications. Every protocol has

significant attributes, for example, the data exchange, reach and power utilization. In view of these attributes such protocols can be ordered in short-range, cellular networks and long-range (Fernández-Ahumada et al, 2019). Conventions for short-range networks (e.g., Bluetooth, ZigBee and Wi-Fi) enable communication of devices in short distances. As per Fernández-Ahumada et al (2019), generally such protocols have a high data transmission rate and low consumption of power. Hence, they are utilized for communication between devices that are close to one another. Protocols for cellular networks (like GPRS, 3G) enable communication in significant distances and with a high data transmission rate. Nonetheless, they have a higher power consumption (Mekki et al, 2019) and high-power consumption (Sanchez-Iborra et al, 2016). Protocols for long range networks (like LoRaWAN and Sigfox) enable communication in extremely significant distances (Sanchez-Iborra et al, 2016). These protocols are utilized to build up the low power wide area networks (LPWAN) because of the way that they have a low power consumption. In any case, the rate of data transmission of these protocols is low. Subsequently, these protocols are fitting for use when the solution requires to send a few amounts of data in extremely significant distances (Sanchez-Iborra et al, 2016). Table 1 presents the attributes of some network technologies utilized for IoT.

Table 2.2: Examples of Network Topology Used in IoT

Parameter	**Wi-Fi**	**Bluetooth**	**ZigBee**	**LoRa**
Standard	IEEE 802.11 a, b, g, n	802.15.1	802.15.4	802.15.4 g
Frequency	2.4 GHz	2.4 GHz	868/915 MHz, 2.4 GHz	133/868/915 MHz
Data rate	2-54 Mbps	1-24 Mbps	20 – 250 kbps	0.3 – 50 kbps
Range of transmission	20 - 100 m	8 – 10 m	10 – 20 m	>500m
Topology	Star	Star	Tree, star, mesh	Star
Power consumption	High	Medium	Low	Very low
Cost	Low	Low	Low	Low

Source: (Fernández-Ahumada et al, 2019).

As demonstrated in Table 2 there is a compromise between coverage, data rate and consumption of energy. Considering the technologies for star networks introduced in Table 2, it is feasible to see that energy consumption is higher in technologies with a high data rate and short coverage. Then again, LoRa has a small data rate yet an enormous coverage and low power utilization. These inquiries are particularly important while considering farming in light of the fact that agricultural situations frequently have limited or no energy supply and impediments for wireless communication.

Various topologies can be utilized for implementing networks, like tree, star and mesh. Star networks have a focal hub and a few peripheral hubs. The communication in such topology happens as follows: peripheral hubs send information directly to the central hub. The central hub can execute capacities for routing messages and communicating through various network protocols (Navarro, Costa & Pereira, 2020). Tree networks are made out of router hubs and leaf hubs. Such networks can be perceived as a bunch of star networks. Inside each cluster, leaf hubs send messages their father node. In mesh networks, in principle, every hub can be a router with rerouting ability. In this way, messages in mesh networks are directed hop by hop until arriving at the last destination (Navarro, Costa & Pereira, 2020).

Data is shipped off the destination through application protocols, for example, the message queuing telemetry transport (MQTT) (Trilles et al, 2018) or the constraint application protocol (CoAP). MQTT is an open-source messaging convention that enables messaging between constraint devices and in unreliable networks. MQTT runs over TCP/IP or similar protocols (like Bluetooth) (Backman, 2019), which uses MQTT fitting for various IoT solutions. The MQTT convention, which depends on the architecture of publish/subscribe, permits communication between devices to occur in the following way. To begin with, devices distribute messages that are organized in topics on a message merchant. At that point, different devices read these messages by subscribing to pertinent topics on the message broker. These topics permit the organization of messages dependent on classifications, subjects, and so forth (Light, R. MQTT Man Page, 2021). The utilization of MQTT for communication between devices permits low coupling between the devices

that distributes the message and the devices that listen to the messages, the purported "one-to-many" correspondence (Trilles, 2018). Like MQTT, CoAP is a communication protocol streamlined for constraint devices and questionable networks. Nonetheless, CoAP messages are exchanged using User Datagram Protocol (UDP) and the CoAP protocol depends on the client-server architecture. This architecture necessitates that a connection is set up between devices before any messages are sent (Trilles, 2018). Consequently, communication using CoAP works in the following way. To begin with, the device that sends messages has to know the address of every device that is relied upon to get messages. At that point, messages are sent over UDP to the predefined address. Because of the use of UDP, CoAP messages are arranged according to the necessary status of affirmation of receival, for instance, confirmable or non-confirmable (Shelby, Hartke & Bormann, 2021). The CoAP convention doesn't execute a construction of themes for messages. Nonetheless, a comparative methodology can be executed using application programming interface (API). In any case, the use of CoAP makes a high coupling between the devices that sends messages and the devices that is required to get messages, as the communication is "one-to-one" (Shelby, Hartke & Bormann, 2021).

The processing layer has data visualization, storage and processing resources. In this unique situation, enormous data permits distributed storage and parallel data processing, enabling the extraction of data in the briefest conceivable time (Ray, 2018). Such information are utilized as models by AI frameworks—which, as indicated by Navarro, Costa & Pereira (2020), can be perceived as the capacity of a system to work as though it had the thinking capacity of a huma being—and AI—that, as per Shi et al (2019) is an information processing method to distinguish patterns and connection among complex and unrelated data—for the improvement of decision support systems and automation of irrigation systems (Adenugba et al, 2019), monitoring (Li et al, 2019) and detection of disease in crops (Lee, Jiong, Son & Lee, 2019), for instance.

Finally, the application layer includes IoT applications that are supported by the other referenced layers, give the management information to farmers, having the option to deal with the whole production process in the plantations (Navarro, Costa & Pereira, 2020)

2.1.4 Architectural Models for IoT Systems

Buyya et al. (2020) proposed a layered architecture for IoT systems, encompassing device, network, service, and management layers. This model emphasizes modularity, scalability, and security across different application domains.

Zanella et al. (2021) introduced a reference architecture for the Internet of Things, highlighting the need for interoperability and standardized communication protocols. Their model focuses on resource-constrained devices and emphasizes efficient data communication.

Al-Fuqaia et al. (2021) present a three-layered architectural model for secure IoT systems. This model prioritizes security at each layer, incorporating authentication, authorization, and encryption mechanisms.

2.1.4.1 IoT Sensors and Their Security

Almazrouei et al. (2023) discuss various types of IoT sensors and their inherent vulnerabilities. They highlight the need for lightweight cryptographic schemes and secure communication protocols to protect data from unauthorized access and manipulation.

Braghin et al. (2023) proposed a lightweight encryption scheme specifically designed for resource-constrained IoT sensors. Their scheme demonstrates promising results in securing data transmission while maintaining low computational overhead.

Siwakoti et al. (2023) presented a comprehensive survey of security challenges and solutions for IoT sensors. They advocate for a layered approach to security, combining hardware-based and software-based techniques for robust protection.

2.1.4.2 Lightweight Cryptographic Schemes

Sim et al. (2022) introduced Lightweight Secure Hash Algorithm (LSH) designed for resource-constrained devices. This scheme offers efficient hashing capabilities while maintaining a small footprint, making it suitable for IoT applications.

Ramakrishnan (2023) proposed the Lightweight Elliptic Curve Cryptography (LECC) scheme, offering efficient encryption and key exchange for constrained environments. LECC balances security and computational complexity, making it attractive for various IoT scenarios.

Guo & Guo (2023) presented a survey of lightweight cryptographic primitives suited for IoT sensor networks. They analysed various schemes based on performance and security metrics, aiding developers in choosing appropriate options for specific applications.

2.1.4.3 Proposed Architectural Model for IoT System

This proposed architectural model builds upon the concepts presented in the literature review and aims to address the specific needs of a project that sends IoT data securely across user and server devices. The model adheres to a layered approach, with each layer handling distinct functionalities:

Perception Layer

- Includes various IoT sensors responsible for collecting data from the physical environment (e.g., temperature, humidity, pressure).
- Employs lightweight cryptographic schemes for data integrity and confidentiality, balancing security with resource constraints of sensors.

Network Layer

- Utilizes reliable communication protocols (e.g., LoRaWAN, MQTT) for efficient data transmission between sensors and the gateway.

- Implements access control mechanisms to restrict unauthorized communication attempts.

Processing and Management Layer

- Processes and analyses collected data using appropriate algorithms and tools.
- Integrates with application-specific systems for further data analysis and utilization.
- Implements user authentication and authorization mechanisms for secure access to data and system functionalities.

Security Layer

- Employs robust encryption techniques (e.g., AES-128) to protect data at rest and in transit within the system.
- Implements intrusion detection and prevention systems to actively monitor for potential security threats.
- Provides secure key management procedures for secure generation, storage, and revocation of cryptographic keys.

2.1.4.4 Summary of Empirical Studies

Several empirical studies have investigated the effectiveness of various architectural models and security approaches for IoT systems. These studies highlight:

- The importance of layered architectures: Modular design allows for easier integration of security mechanisms and facilitates scalability for large-scale deployments.
- The need for lightweight cryptography: Resource-constrained devices require efficient cryptographic schemes to ensure minimal impact on performance and energy consumption.

- The trade-off between security and performance: Finding the optimal balance between robust security and efficient resource utilization is crucial for successful IoT system implementation.

Future research should focus on:

- Developing and evaluating novel lightweight cryptographic schemes: Tailored to specific application domains and resource constraints of different IoT devices.
- Adapting existing architectural models to address emerging IoT applications: Emphasizing interoperability and scalability to cater to the diverse and evolving needs of IoT systems.
- Conducting more empirical studies: Evaluating the effectiveness of proposed models and security solutions in real-world scenarios.

By building upon existing knowledge and conducting further research, we can design and implement robust and secure architectural models for diverse IoT applications, fostering their successful adoption and utilization across various domains.

2.2 IoT in Sub-Saharan Africa

The sub-Saharan Africa greatly relies on agriculture for survival. A report by Fusi & Mbarika, (2022) report that 32% of sub-Saharan Africa's GDP is generated due to agriculture, and the sector has employed more than half of the entire labor force. Scholars and startups are proposing several IoT platforms to suit the Sub-Saharan Africa. The proposals are of systems that include sensors for soil state like temperature, moisture, nutrients and pH, and provide data on water and microclimate conditions. Data processing and storing is provided locally (Fusi & Mbarika, 2022).

Research by Negussie *et al* 2020) focuses in DeKUT farm, which at first settled as a plantation yet has advanced over the long run to give top notch facilities, assets and openings for research in yields and animals. It has 650 acres of land of homestead of which 400 acres of land are under harvest and vegetable creation while the excess segment is

held for biodiversity shelter focuses. The harvests and vegetables in the farm are coffee (294 acres of land), maize (45 acres of land), bananas, beans, arrowroots and a combination of vegetables (36 acres of land). Coffee and 14 acres of land of vegetables are under irrigation. The coffee partition is watered through overhead while vegetables are under drip irrigation system. A petroleum-controlled pump is used for conveying water to watering hosepipes in the drip irrigation system. The farm additionally has three medium-sized greenhouses of which two of them are drip irrigated. Water for irrigation system is collected from a close by river Muringato. The Farm has a rain gauge measure and keeps an everyday rainfall record with a normal rainfall of 26 mm during stormy season. There has been continuous clogging up of water emitters prompting their dysfunction with extra maintenance cost (requirement of water filters and regular flushing of pipe systems). The irrigation system timetable and water amount are the same paying little mind to individual plant's requirements, which influences the yield and causes loss of water. High gifted agricultural expert chooses the timetable of water system for the farm. To decrease the expense of difficult work, actuator for opening/shutting the emitters is required. These issues can be tended to through the incorporation of IoT frameworks that comprises different sensors, actuators and irrigation scheduling framework. The IoT framework considers the particular necessities of each crop at its distinctive developing stage, the encompassing weather conditions, real-time soil parameters and soil properties to settle on scheduling decisions. (Negussie et al, 2020)

The knowledge and deployment of IoT systems in agriculture shall benefit Africa towards achieving zero hunger, according to Iorliam, Iorliam & Bum (2020). The authors recommend that government institutions, agricultural entrepreneurs, farmers and agricultural research institutes should leverage on the benefits of IoT systems in agriculture to increase crop production, monitor farms in real time remotely, monitor agricultural equipment which improves cost of operation and improve livestock and crop farming. Application of IoT in agriculture is a technology that is already utilized in America, Asia and Europe to solve their agricultural challenges. If African nations adopt these techniques with their large tracks of land, water supply from rivers and rain

reservoirs, poultry and livestock, there will be high productivity hence achieving zero hunger target as well as optimize energy usage in farms (Iorliam, Iorliam & Bum, 2020).

To take care of a developing worldwide population projected to increment to 9 billion by 2050, production of food should increase from its present level. The majority of this growth should come from smallholder farmers who depend on generational knowledge in their cultivating practices and who live in areas where seasons and weather patterns are getting less predictable because of environmental change. The expansion of IoT is increasing the freedom to apply digital devices and services on smallholder farms, which includes soil and plants monitoring for agriculture, water quality in hydroponics, and ambient conditions in nurseries. In blend with other food security efforts, IoT-enabled precision for smallholder farming can possibly improve livelihood and speed up low and middle-income nations' journey to self-reliance (Antony *et al*, 2020).

According to Fastellini & Schillaci (2020), smart farming in Africa was impeded by advancement of ICT systems around the continent. Recent projects, like the Last Mile Project and other projects that interface sub-Saharan Africa to the world Internet have sparked researchers and different partners in agribusiness to take an action on digitization of agriculture and smart farming. These endeavors will go so high when Internet costs goes down and gets accessible to the greater most farmers. Farmers in Africa are ready to receive and adopt innovation that would improve on their day-by-day economic activities. That is the reason why mobile money in phones is so well known in Africa and helpful among farmers.

According to BRCK, about 800 million Africans do not have internet access making internet coverage, especially in rural areas, low (brck.com). As much as the acquisition of phones is growing in Africa, smartphone adoption is modest. Common types of mobile devices owned are basic phones. Worldwide, sub-Saharan Africans report the lowest ownership rate of smartphones compared to any other geographic region. Additionally, according to M-Kopa (https://m-kopa.com/impact/), one of the leading solar home system states that 75% of sub-Saharan Africa remains unconnected to the internet. Sending IoT

data can use general packet radio service (GPRS), which is an advancement of GSM. However, since GPRS is not dedicated for transmission of IoT data due to weaknesses in power efficiency and coverage, GSM is preferred (Bima, Suryani & Wardana, 2020).

2.3 Threat Models in IoT

A. Attacks against Privacy

Yang, Derhab & Maglaras (2020) shows that the classification of attacks depends on learning the exact area and character of IoT devices at agriculture sensors to get private information and compromise the security of the system. In green IoT-based horticulture, the IoT information (e.g., the water composition, temperature, and humidity) is gathered on numerous occasions each hour by IoT devices and intelligent meters to acquire fine-grained data about the plants status and improve supplement solution efficiency. The detailed examination of this IoT information may effortlessly uncover farms' proactive tasks and the adopted nutrient solution. For instance, in pH settings, if the pH rises exorbitantly shows that the farmer will increase the ammonium supply, and if the pH falls demonstrates that the farmer will diminish the ammonium supply. Utilizing this data, an attacker can design actual attacks (e.g., sending a drone) to disturb pH settings. Clearly, this private data (for example pH settings) should be shielded from unapproved access (Yang, Derhab & Maglaras, 2020).

B. Attacks against Authentication

This classification of attacks fashions characters to imitate as approved nodes (i.e., IoT gadget, cloud node, or fog node) to access the green IoT-based agribusiness. For instance, an enemy may launch the accompanying identity-based attacks for forged identities, to be specific, replay attacks, masquerade attack, impersonation attack, and spoofing attack (Yang, Derhab & Maglaras, 2020).

A replay attack happens as man-in-the-middle attack (MITM). Its targets in the green IoT-based farming are to capturing data packets between IoT gadgets or an IoT gadget with a

access point at agricultural sensors section and afterward transferring them to their objections without alteration. The authentication protocols for securing IoT networks utilize three strategies against replay assaults, in particular, paring-based cryptography, hash functions, and timestamp in the encoded information, as talked about by Ferrag et al (2017) in their previous paper.

A masquerade attack intends to take on the appearance of a real node to sign into the worker at farming sensors layer (i.e., sign into the access point) or fog computing (i.e., sign into the fog node). The authentication protocols that secure the IoT networks utilize three techniques against masquerade attack, in particular, 1) social highlights based biometric (e.g., keystroke, mark, walk, or voice), 2) behavioral-based biometric (e.g., unique finger impression palm, electrocardiogram, eyes, or face), 3) hashing, 4) Elliptic curve cryptosystem, and 5) matching-based cryptography Ferrag, Maglaras & Derhab, 2019).

C. Attack against Confidentiality

This class of attacks endeavors to adversarial snoop the network traffic between IoT devices or an IoT device with an access point at the sensors to misdirect the green IoT-based horticulture to bargain the confidentiality and settle on wrong actions/decision. For instance, an attacker may launch the accompanying Eavesdropping-based attacks to compromise the classification, including, tracing attack, known-key attack and brute force attack (Yang, Derhab & Maglaras, 2020).

A tracing attack aims to gather sufficient security data from IoT devices at agriculture sensors to interface information with a specific genuine personality. To counter this attack, security arrangements dependent on random numbers in commitments and confirmations should be created (Guo et al, 2013).

A brute force attack plans to create a rundown of all potential passwords that can be utilized by IoT devices at farming sensors, at that point to debilitate them individually until the right secret phrase can be distinguished (Salamatian et al, 2019).

A known-key attack plans to produce new session keys dependent on trading off past meeting keys. To resist this attack, security protocols that coordinate random nonce in meeting key should be created, like in the case of JSON Web Token and JSON Web Encryption (JWT and JWE) (Yang, Derhab & Maglaras, 2020).

D. Attacks against Availability

This classification appears as Denial of Service (DoS) attacks. Its will probably make the services in green IoT-based agriculture (e.g., validation for IoT gadgets) are inaccessible either by (1) flooding servers with an immense measure of information to make it occupied and unfit to offer an assistance to IoT gadgets; (2) refreshing with false data injection attacks; or (3) attack on exact localization for UAV with a noxious 5G station (Yang, Derhab & Maglaras, 2020).

E. Attack against Integrity

This class of attack infers an unapproved party to getting to and altering private data (for example pH settings). Under this class, we can track down the accompanying attack: forgery, man-in-the-middle (MITM), biometric template attack, and trojan horse. To oppose this assault, the data aggregation schemes dependent on homomorphic encryption and hash functions should be created Ferrag, Maglaras & Derhab, 2019).

2.4 Data Authentication and Authorization Techniques

2.4.1 JWT

McCarthy et al. (2015) say that JWT is a set of base64-encoded JSON objects that are digitally signed and allow stateless REST-based frameworks to manage sessions and

claims. A JSON-based open standard for access token creation is called JSON Web Token (JWT). A server might create a token with the claim "logged in as admin" and give it to a client as an illustration. The client might then utilize that token to demonstrate that it has an administrator login. Since the tokens are signed using the server's key, both the client and the server can confirm that the token is authentic. The tokens are intended to be small, safe for URLs, and usable, particularly in single sign-on (SSO) scenarios for web browsers. JWT claims, or any other sort of claims as required by systems and processes, can be used to transfer the identity of authorized users between an identity provider and a service provider. The tokens may also be encrypted and authenticated (Haekal, 2016).

JWTs typically consist of three components: a signature, a payload, and a header. The header, which appears like this: indicates the algorithm used to create the signature.

```
header = '{"alg":"HS256","typ":"JWT"}'
```

HS256 indicates that this token is signed using HMAC-SHA256.
The payload contains the claims to make:

```
payload = '{"loggedInAs":"admin","iat":1422779638}'
```

As suggested in the JWT spec, a timestamp called iat is installed.
The header and payload are base64url encoded and concatenated with a period as a separator to create the signature:

```
key          = 'secretkey'
unsignedToken = encodeBase64(header) + '.' + encodeBase64(payload)
signature    = HMAC-SHA256(key, unsignedToken)
```

The signature is base64url encoded to bring everything together. Periods are used to join the three distinct portions:

token = encodeBase64(header) + '.' + encodeBase64(payload) + '.' + encodeBase64(signature) # token is now: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJsb2dnZWRJbkFzIjoiYWRtaW4iLCJpYXQiOjE0MjI3Nzk2Mzh9.gzSraSYS8EXBxLN_oWnFSRgCzcmJmMjLiuyu5CSpyHI

In contrast to XML-based standards like SAML, the result is three Base64url strings divided by dots, which may be simply given in HTML and HTTP settings. The two most common cryptographic techniques are RSA signature with SHA-256 and HMAC with SHA-256 (HS256) (RS256). Many more are introduced in the JWA (JSON Web Algorithms) RFC 7518 for both encryption and authentication. The JOSE header refers to the first portion of the JWT (after the periods). The IETF working group known as JOSE—short for Javascript Object Signing and Encryption—works to standardize the encoding of integrity-protected data using JSON data structures. It's a signed message, according to the JOSE header above. Google validates the end-identity of the user by signing the JWT, which contains data about the user's identity (Haekal, 2016).

2.4.2 JWS (JSON Web Signature)

A signed JWT is referred to as a JWS (JSON Web Signature). In fact, a JWT does not exist in and of itself; it must be either a JWS or a JWE (JSON Web Encryption). The JWS and JWE are concrete implementations of an abstract class. The JWS specification is not tied to any particular algorithm. All applicable signing algorithms are defined in the RFC 7518 JSON Web Algorithms (JWA) specification. RFC 7518 section 3.1 defines all possible alg(algorithm) element values for a JWS token. The value of the kid element indicates or hints about the key that is used to sign the message. Looking at the ***kid***, the message's recipient should know where and how to look for and find the key. Members of the JSON object represented by the JOSE header in a JWT describe the cryptographic operations applied to the JWT and, optionally, additional JWT properties. The rules for JOSE header values differ depending on whether the JWT is a JWS or

JWE. The JOSE header is required under the JWS and JWE — in other words, there is no JWT without a JOSE header.

JOSE header

JWTs that have been signed and encrypted include a header known as the JOSE header (JSON Object Signing and Encryption). This header specifies which algorithm (signing or encryption) is used to process the JWT data. Typically, the JOSE header defines two attributes: `alg` and `typ`.

- alg: the algorithm utilized in signing or encrypting the JWT
- typ: the content being signed or encrypted (typically 'JWT') (Ethelbert *et al*, 2017).

Compact Representation

In addition, JWS defines a compact representation for a signed JWT:

BASE64URL (UTF8(JWS Protected Header)) + '.' +

BASE64URL (JWS Payload) + '.' +

BASE64URL (JWS Signature)

The compact representation is essentially the concatenation of the JOSE header, the JWT, and the signature details. Each component is encoded in BASE64 and separated by a single dot ('.').

This yields the typical JWT representation found on the web:

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

Compact Representation for Encrypted JWTs

The compact representation for encrypted JWTs differs slightly:

BASE64URL (UTF8(JWE Protected Header)) + '.' +
BASE64URL (JWE Encrypted Key) + '.' +
BASE64URL (JWE Initialization Vector) + '.' +
BASE64URL (JWE Ciphertext) + '.' +
BASE64URL (JWE Authentication Tag)

A JWT would normally be included in the ciphertext.
JWTs that have been signed and encrypted are typically nested. That is, a signed JWT is created first, followed by an encrypted version of the signed result. This has two advantages:

- The signature cannot be removed.
- The signature is private (no one else can see it). (Hunt, Deniss& Ansari, 2018).

Common JWT Signing Algorithms

The majority of JWTs in the wild are simply signed. The most common algorithms are as follows:

- HMAC + SHA256
- RSASSA-PKCS1-v1_5 + SHA256
- ECDSA + P-256 + SHA256 (Hunt, Deniss& Ansari, 2018).

a. HMAC algorithms

This is most likely the most commonly used algorithm for signed JWTs.

Hash-Based Message Authentication Codes (HMACs) are a class of algorithms that allow messages to be signed using a shared key. A cryptographic hash function is used in the

case of HMACs (for instance SHA256). The strength (i.e. how difficult it is to forge an HMAC) is determined by the hashing algorithm used.

The algorithm's main goal in design was to allow the combination of a key and a message while providing strong guarantees against tampering. Ad hoc solutions (such as appending the key to the message and hashing the result) have mathematical flaws that allow potential attackers to forge the signature. The HMAC algorithm is intended to prevent this.

The algorithm itself is quite straightforward (JavaScript pseudo-code with Node.js extensions):

```
// Key: Buffer with key, Message: Buffer with message
functionhmacSha256(key, message) {
// The algorithm requires the key to be of the same length as the
// "block-size" of the hashing algorithm (SHA256 = 64-byte blocks).
// Extension is performed by appending zeros.
varfullLengthKey = extendOrTruncateKey(key);
varoutterKeyPad = 0x5c; // A constant defined by the spec.
varinnerKeyPad = 0x36; // Another constant defined by the spec.
varoutterKey = new Buffer (fullLengthKey.length);
varinnerKey = new Buffer (fullLengthKey.length);
for (var i = 0; i <fullLengthKey.length; ++i) {
outterKey[i] = outterKeyPad ^ fullLengthKey[i];
innerKey[i] = innerKeyPad ^ fullLengthKey[i];
   }
// sha256(outterKey + sha256(innerKey, message))
// (Buffer.concat makes this harder to read)
return sha256(Buffer.concat([outterKey, sha256(Buffer.concat([innerKey, message]))]));
}
```

HMACs are used with JWTs when all parties need a simple way to create and validate JWTs. Anyone with the key can generate new JWTs. In other words, using shared keys, one party can impersonate another: HMAC JWTs do not provide any guarantees about the creator of the JWT. Anyone with the key can make one. This is overly permissive in some cases. Asymmetric algorithms come into play here (Zhang, Alger & Bucher, 2018).

a. RSA and ECDSA algorithms

RSA and ECDSA are asymmetric encryption and digital signature algorithms, respectively. Asymmetric algorithms provide the capability of verifying or decrypting a message without the possibility of creating a new one. This is critical in some cases. Consider a huge company where data generated by the sales team must be verified by the accounting team. If an HMAC is used to sign the data, both the sales and accounting teams must have access to the same key. This would enable the sales team to sign data and have it pass through as if it came from the accounting team. Although this may appear unlikely, particularly in the context of a corporation, there are times when knowing who created a signature is critical. This capability is provided by JWTs signed or encrypted with RSA or ECDSA. A JWT is signed by a party using its private party. Receivers then use that party's public key (which must be shared in the same way as an HMAC shared key) to verify the JWT. Receiving parties are not permitted to generate new JWTs using the sender's public key.

The fundamental distinction between RSA and ECDSA is in speed and key size. Both the RSA and ECDSA algorithms are more sophisticated than HMAC. To attain the same level of security as RSA, ECDSA needs smaller keys. This makes it an excellent option for small JWTs. But typically, RSA is quicker than ECDSA. As always, choose the option that most closely matches your needs.

b. Lightweight Service

REST services are designed to be lightweight. The type of service that is used to build new architectures typically includes some or all of these features (Zang, Alger and Bucher, 2018):

- HTTP transfer. All cloud platforms support it because it is inexpensive and simple to deploy.
- Message content that is text-based, typically JSON. The emphasis is on readability and, on occasion, on providing simple support to front-end browser and mobile developers.
- Brief messages and prompt responses. Although there is a strong parallel trend of services with streaming responses for "push" data, they are not always implemented in the same way, and HTTP may not be the best transport in those cases.
- REST-like, or inspired by the REST architectural style.
- Some statelessness, either 'share-nothing' or very careful use of server-side state, to allow for horizontal scalability.
- Interoperability. The service's users could be a dynamic population, completely unknown or unknowable at the time of design, and using a variety of platforms and languages. (Zang, Alger and Bucher, 2018).

What Are the Security Requirements for a Lightweight Service?

It is possible, but unlikely, that a service is so light that no security is required, such as if it is purely informational (Silva, Leal &Paiva, 2018). If security is required, it will be for the usual reasons, namely identity and permission; knowing who is requesting a resource and determining whether they are authorized to use it. Identity and permission are frequently referred to as "authentication" and "authorization" (particularly in Spring Security), but using those terms would be confusing because "authorization" is part of the OAuth2 domain language and has a different meaning there. "Authorization" is also the

name of a standard HTTP header, which is slightly different in that it is concerned with data transport rather than permission calculation.

So, the fundamental requirements are identity and permissions, and the options all boil down to the following details:

- How is identity and permission data communicated to a service?
- How is this data encoded and interpreted?
- What information is required to make the access decision (user accounts, roles, ACLs, and so on)?
- How is data managed, and who is in charge of storing and retrieving it?

The remainder of this article addresses those questions using a couple of example approaches, including OAuth2, but does not purport to be a comprehensive description of available features and implementations (Silva, Leal &Paiva, 2018).

2.4.3 Enhancing IoT Communication: A Deeper Look at JWT and JWS

The Internet of Things (IoT) is rapidly transforming various sectors, enabling communication between diverse devices and systems. However, ensuring secure and reliable communication in this complex network is crucial. Two promising technologies, JSON Web Token (JWT) and JSON Web Signature (JWS), can significantly enhance communication security and efficiency in the context of IoT (De Diego et al, 2021).

Understanding JWT and JWS

JWT (JSON Web Token) is a compact and self-contained way of securely transmitting information between parties as a JSON object. It encompasses three parts:

- Header: Contains metadata about the token, such as the signing algorithm used.

- Payload: Carries the claims, which are statements about the subject (e.g., device identity) and other relevant information.
- Signature: Ensures data integrity and authenticity by being generated with a cryptographic key.

JWS (JSON Web Signature): is a subset of JWT that focuses solely on the signature aspect. It utilizes a digital signature to verify the integrity of the data within the payload and the identity of the party that created the JWS (De Diego et al, 2021).

Benefits of JWT and JWS in IoT Communication

Security: JWS provides data integrity and authenticity through digital signatures, preventing unauthorized modifications and ensuring data originates from a trusted source. JWT can optionally contain claims regarding access control, allowing authorization decisions at the receiving end without requiring centralized servers.

Scalability: Both JWT and JWS are compact and lightweight, minimizing bandwidth consumption, which is crucial for resource-constrained IoT devices. The stateless nature of JWT reduces server load and improves scalability in large-scale IoT deployments.

Interoperability: Both technologies adhere to widely recognized JSON and cryptographic standards, facilitating seamless communication between different devices and systems regardless of vendor or platform (Xu et al, 2023).

Considerations for Implementation

Key Management: Secure key management strategies are crucial to ensure the effectiveness of both JWT and JWS. This includes secure storage and access control mechanisms for cryptographic keys.

Device Capabilities: Resource limitations of IoT devices should be considered when implementing these technologies. Lightweight signing algorithms and efficient token verification processes are crucial for smooth operation.

Standardization: Compliance with established standards (e.g., RFC 7519 for JWT) ensures interoperability and facilitates integration with existing infrastructure (Xu et al, 2023).

JWT and JWS offer valuable tools for enhancing security, scalability, and interoperability in IoT communication. By understanding their functionalities and addressing implementation considerations, these technologies can contribute significantly to building a robust and secure communication infrastructure for the ever-evolving world of IoT (Xu et al, 2023)..

2.4.4 Related Studies

Recently, Waghmare et al., (2020) integrated IoT and GSM with a transformer to monitor and protect it. Their system uses a microcontroller and is monitored using GSM technology. The microcontroller gets initialized, and C code is compiled and uploaded to the Arduino board from a USB port. The transformer values are then compared and checked against the preset values that are fed in the Arduino. Examples of these values include voltage where, if an overvoltage occurs, its relay starts to operate. Its values are displayed on an LCD screen, and messaging is done using GSM technology. In the presence of any abnormality, transformer details are updated on a webpage, and an alert text message is sent. Although the system utilizes a GSM communication network with low investment and operation costs, no data protection or security of the transformer is proposed. If any, it is undisclosed.

Antonio et al., (2020) propose an IoT platform for the benefit of rural African farmers. The platform has an IoT sensing platform that provides the state of the soil information like moisture, light, air temperature, color, and texture. The system’s hardware is housed in a specially designed casing for easy assemblage and protection, deployment, and

transport. While the solution targets rural farmers, there isn't any security of data implemented to ensure that the user data is not acquired maliciously.

Sigu et al., (2020), in collaboration with the International Cancer Institute, are utilizing IoT to support oncology clinics in the rural areas of Kenya. Cancer patients get the information needed from the physician without going to the clinic facilities. Patients are monitored, and access information in real-time reduces costs and improves the patient outcome of treatment. Patients are also trained through an online module. Remote monitoring is beneficial to prevent lengthy hospital durations and readmissions. The use of technology has proved to have a significant impact on cancer patients in rural sub-Saharan Africa. The application of this technology, however, lays minimum emphasis on the security of sensitive data of patients. No frameworks or information about how the patients' data security has been mentioned.

With the era of the use of informational renovation to enhance performance in water pipelines, Priyanca et al., (2020) proposed integration of Internet of Things to monitor pressure, viscosity, pumping station parameters and other external parameters. The paper shows how the Internet of Things has progressed the industrial element of monitoring and intellectual control of the pipeline systems. The IoT system proposed is a smart module to enhance efficient data communication. The system module however, isn't clear on its security of its data, if any.

2.5 Knowledge Gap

From the literature review, there are a lot of issues involved with adoption and security of IoT in sub-Saharan Africa.

Despite limited resources and infrastructure, several noteworthy IoT projects are emerging across Sub-Saharan Africa. These projects span various sectors, including:

- Agriculture**:** Precision agriculture solutions utilizing sensors and data analytics in Kenya (Asare, 2023) and Tanzania (Ewan, 2023) to optimize irrigation, monitor crop health, and improve yields.
- Healthcare**:** Remote patient monitoring systems in South Africa (Vodacom, 2023) enabling early detection and management of chronic diseases in Rwanda (Parsa, 2023).
- Environment**:** Wildlife tracking and monitoring systems to combat poaching and protect endangered species (African Wildlife Foundation, 2023).
- Smart Cities: Traffic management and infrastructure monitoring systems for improved efficiency and resource utilization in Rwanda (JICA, 2023) and Ghana (Eduam, 2023).

These projects demonstrate the potential of IoT to address various challenges and improve lives in Sub-Saharan Africa. However, several challenges hinder widespread adoption and successful implementation:

- Limited access to electricity and internet: Extensive areas lack reliable power and internet connectivity, crucial for powering and connecting IoT devices (The Ocean Cleanup, 2023).
- Lack of technical expertise: Skilled personnel for deployment, maintenance, and data analysis are often scarce, creating operational hurdles (International Labour Organization, 2020).
- High cost of implementation: Setting up and maintaining IoT infrastructure can be expensive, posing a barrier for resource-constrained communities (GSMA, 2023).

Data Security and User Privacy Concerns

As with any technology involving data collection and transmission, data security and user privacy are significant concerns in the context of Sub-Saharan Africa's IoT landscape. These concerns include:

- Data breaches and leaks: Inadequate security measures can expose sensitive data collected by IoT devices, leading to privacy violations and potential misuse (Skouloiudi, 2020).
- Lack of awareness and regulation: Limited understanding of data privacy rights and weak regulatory frameworks can leave users vulnerable to exploitation (UNECA, 2023).

To address these concerns, several solutions can be explored:

- Implementing robust data security measures: Encryption, secure communication protocols, and user authentication are crucial for protecting sensitive data (Lukitowati & Ramli, 2020).
- Raising awareness and education: Educating users about data privacy rights and responsible data use is essential for building trust and promoting responsible practices (Mkomwa et. al, 2022).
- Developing and enforcing data privacy regulations: Establishing clear regulations and frameworks for data collection, storage, and usage is crucial for protecting user privacy (Mkomwa et al, 2022).

Integrating IoT with Existing Infrastructure

To maximize the impact and ensure efficient resource utilization, it is crucial to explore how proposed IoT solutions can integrate and enhance existing infrastructure. This can be achieved through:

- Leveraging existing communication networks: Utilizing existing mobile networks or low-power wide-area networks (LPWANs) can provide cost-effective and efficient connectivity solutions (GSMA, 2021).
- Building upon existing platforms: Integrating proposed IoT solutions with existing platforms, such as e-government or healthcare management systems, can

streamline data management and facilitate information exchange (Kuteyi & Winkler, 2022).

- Prioritizing interoperability: Ensuring compatibility between different devices and systems is crucial for seamless integration and data exchange (ITU, 2020).

By focusing on these approaches, proposed IoT solutions can leverage existing infrastructure to create a more robust and sustainable ecosystem, maximizing their positive impact on Sub-Saharan Africa.

Summary

A literature has been conducted on IoT acquisition in farming in the sub-Saharan Africa. Various techniques that are currently being used have been noted, as well as vulnerabilities that pose a risk to IoT systems. From the research gap, IoT can be used to increase food security in Africa, but due GPS coverage and security problems, the acquisition of these systems are slowed down. This thesis looks at multimodal technique for securely sending and receiving data from IoT systems using GSM in sub-Saharan Africa, to address the issues brought about in the literature.

CHAPTER THREE

METHODOLOGY

3.1 Introduction

This chapter gives a detailed research method to ensure quality work. It gives more detailed methods and techniques to be used when collecting data and the analysis of the MAAMSIC model. A comparative analysis was also be made.

3.2 Research Design

Research design is the overall plan for connecting the conceptual research problems to the pertinent and achievable in the research. In other words, the research design articulates what data was required, what methods were used to collect and analyze this data, and how all of this was going to answer our research question. Enhance the security by the use Multimodal Authentication and Authorization Model for the Security of IoT Communication via GSM Messaging (MAAMSIC) was qualitative research method. This method was associated with inductive approaches (based on empirical evidence).

The purpose of the research was to collect data of both authentication techniques from literature review then summarize them for new design recommendation. Thus, the collections of secondary data provided necessary information to design new multimodal authentication technique. The researcher's role was to analyze and design how to construct Multimodal Authentication and Authorization Model for the Security of IoT Communication via GSM Messaging (MAAMSIC). Various techniques including documentation, observation and analysis of both authentication techniques (MAAMSIC and phishing app) were used. There was the use of empirical methodology by evaluating functionalities level of the authentication techniques to achieve high level of security. The design of MAAMSIC used PHP Framework called Yii, Flutter (Dart) and IDE known as Visual Studio Code, SMS Gateway and IoT hardware that has moisture sensor, GSM chip and water level sensor. Flutter was used to create the interface where the user interacts

and tests data, whereas Yii was used to create server, SMS Gateway API and IoT Communication module. Evaluation of the Multimodal RESTful Authentication model was done using the registered and unregistered phone numbers, scenarios that are based on real life to get unauthorized data. The implication of new features proven was also described to respond research objectives

3.2.1 Methods

To be able to carry out research, there are many research methods and data collection techniques available to follow in order to achieve the research finding. However, the selection of research method depends on problem at hand. There are two various types of research methods / research techniques available for data gathering and analysis: Qualitative Research, Experimental, Survey, Delphi Study, Case Study Method and Quantitative Research. Experimental Method was appropriate in this study, since it was a systematic and scientific method approach to research. With this method, we reviewed the variables of security, latency and resource utilization in the Multimodal RESTful Authentication model and manipulated, these components to investigate with previous models. The Experiment method enabled finding out important features of designing the new Multimodal Authentication and Authorization Model for the Security of IoT Communication via GSM Messaging (MAAMSIC).

3.2.2 Tools

A software program was written to demonstrate the effectiveness and feasibility of the technique using Yii, PHP framework. Visual Studio Code was the software development platform used. It allowed applications to be developed from a set of modular software components called modules. Visual Studio Code was used to perform coding of mobile application, server code and SMS Gateway API, the one used to interact with the IoT device. The IoT device used was an IoT simulator called Iotify, which showcased a real representation of an IoT device.

3.3 Population and Sample

The selection criteria for this study focused on testing the security of the MAAMSIC app in preventing unauthorized data access from IoT devices in Sub-Saharan Africa. This aligns with the research finding that security concerns are a significant barrier to IoT adoption in the region. The target population for this investigation was IoT devices commonly used in Sub-Saharan Africa (i.e., smart hospital devices, water pressure sensors, agricultural sensors). However, due to resource limitations, the study employed a sample of 4 specific IoT devices in the sector of health and agriculture. These devices were chosen based on their popularity and representation of the broader target population.

Population Sample

A convenience sampling technique was utilized, where readily accessible devices were selected. This approach acknowledges limitations in obtaining a truly representative sample but is justified due to the initial exploratory nature of the study. Future research could employ more robust sampling methods, such as stratified or random sampling, to achieve greater generalizability.

The sample size of 5 devices was chosen based on resource constraints and the need to balance data collection feasibility with obtaining sufficient data for meaningful analysis. Additionally, considering the exploratory nature of the study, a smaller sample size was deemed appropriate for initial testing purposes.

Rationale for Sample

This study focused on registered and unregistered phone numbers, as these represent potential avenues for unauthorized communication attempts. By testing both types of numbers, the research aimed to evaluate the MAAMSIC app's effectiveness in mitigating security risks regardless of the sender's identity on both the server side and user end.

3.4 Research and Data collection

3.4.1 System Development and Testing

This section describes the development and testing methodology employed for the MAAMSIC system and the phishing app.

A multimodal system was created to emulate the typical development process for IoT systems. This system was then tested with both registered (and verified) and unregistered phone numbers. The interaction with the system generated secondary data that was subsequently collected and analysed. This data provided valuable insights for the implementation of the proposed Multimodal RESTful Authentication model.

Three key parameters were prioritized during testing:

1. **Performance/Speed:** Evaluating the system's responsiveness and processing efficiency.
2. **Security:** Assessing the system's ability to safeguard data from unauthorized access.
3. **Signature Creation:** Examining the effectiveness of the chosen signature method.

3.4.2 Data Collected

To ensure comprehensive testing and model development, the following data points were collected during interaction with both the MAAMSIC system and the phishing app:

Table 3.1: Data Collection for MAAMSIC

Data Category	Description
Request/Response Time	Time taken for the system to process and respond to requests.
Authentication Attempts	Number of attempts made to access the system, successful and unsuccessful.
Security Events	Any detected unauthorized access attempts or security breaches.
Signature Verification Results	Outcome of the system's verification of the attached signature for each request.
User Interaction Data	User actions within the system, including navigation patterns and specific functions used. (Note: This data collection should be done ethically and with user consent)

3.5 Data Analysis

The analysis conducted on above secondary data leads to the method uses JWT and JWS for authentication process.

The encryption process ended up making the final Multimodal RESTful Authentication model more difficult to be broken using existing authentication in order to achieve highly security of confidentiality integrity and authenticity

3.6 Model and Algorithm Development

The parameter estimated value from statistical distribution model was used to construct the Multimodal RESTful Authentication and authorization model in this research. For ease of interaction with the model an algorithm was created to describe the operation of the model. The model provided a description of how to evaluate the individual predictive components of the model.

For ease of interaction with the model, an Auth portal and 2 mobile apps were created for ease of interaction within the software developers' environment. The portal provided a Graphical User Interface for data entry of data inputs and reports of final display of data.

Additionally, to cater for users who don't have a smart phone, an SMS channel was created for them to send and receive commands from the IoT through the secured server

3.7 Model Validation

Multimodal Authentication and Authorization: Traditional authentication approaches often rely on a single factor, such as passwords or tokens. Multimodal authentication strengthens security by requiring users to provide multiple types of credentials, such as something they know (password, PIN), something they have (smartphone), or something they are (fingerprint). This layered approach makes it significantly more difficult for unauthorized individuals to gain access to protected data.

Model Testing and Validation

To assess the effectiveness of the proposed Multimodal RESTful Authentication model in enhancing web resource security, validation tests were conducted. The model, detailed in Figure 1, was tested by attempting to access critical data using both smartphones and simple phones. This diverse testing environment reflected the varying technological access levels prevalent in Sub-Saharan Africa.

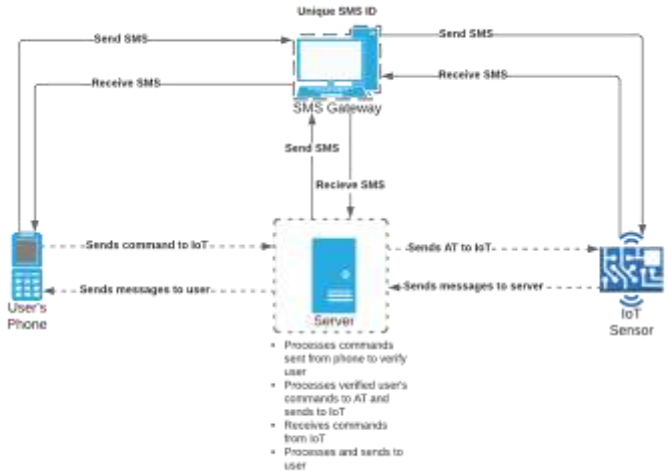

Figure 3.1: MAAMSIC Architecture

The obtained results, discussed in a subsequent section, were utilized to analyse the model's performance and validate its theoretical underpinnings. By comparing the effectiveness of the model across different device types, the validation process aimed to demonstrate its technological contribution by:

- Enhancing security: Implementing multimodal authentication significantly raises barriers to unauthorized data access compared to traditional single-factor methods.
- Improving accessibility: The model's ability to function on both smartphones and simple phones caters to users in regions with limited access to advanced technology, promoting wider adoption and utilization of IoT systems, especially in rural environments.

3.8 Study Hypotheses

This section investigates the research questions that guided the study:

Hypothesis 1: Current IoT systems often prioritize functionality and user experience over robust security measures, leading to user dissatisfaction and abandonment shortly after acquisition.

Hypothesis 2: By offering the option to utilize both simple phones and smartphones for authentication, the model can address the challenges posed by limited internet coverage and lack of advanced technology access prevalent in many parts of Sub-Saharan Africa. This inclusivity can significantly increase the adoption and utilization of affordable IoT systems, particularly in rural areas.

3.9 Summary

This research was a qualitative based research method. This method was associated with inductive approaches based on observation data tested against the existing models demonstrated by the phishing app, and the secure model demonstrated by the MAAMSIC model. Practical and comparative studies have been done to guarantee the construct of secured Multimodal RESTful Authentication model. The best authentication was the one satisfying the high level of four security elements, confidentiality, integrity, authentication and nonrepudiation. During data collection, processing time and key management are important items in authentication. Functional analysis of the RESTful authentication techniques assists to fulfill security requirements. JWT and JWS had the advantage of securely transporting the generated signature (it faces the challenge of signature generation). Therefore, the smartphone app was encrypted with JWS to eliminate third-part apps from reading the sensitive data, and can securely access the server encrypted with JWS. Since JWT can be decoded by base-64 algorithms, JWS signs it further to ensure that the data cannot be decoded.

CHAPTER FOUR

EXPERIMENT DESIGN, DEVELOPMENT AND IMPLEMENTATION

This chapter focuses on creation of multimodal model that encompasses the model architecture, server app, adding encryption to the model, hardware and software implementation. The chapter further evaluates the model against metrics of latency, security and resource utilization, and compares it to other models for validation.

4.1 System Architecture

The system is made up of three major modules:

An IoT sensing platform equipped with a soil moisture sensor detects moisture content and sends signals to begin irrigation of crops. The device's signals are routed through the GSM network. To monitor the amount of water in the irrigation tank, a water level sensor is installed. The server is linked to the water level sensor. This is to ensure that users can send requests to the SMS Gateway from their phones to monitor the water level.

PHP is used to write the server software, which is hosted on a server. The server is linked to an SMS gateway provider in order to receive messages from the device and relay them to the user. Before sending and receiving commands, the server verifies that the user is registered and has a verified phone number and device. The server also ensures that users are authenticated and that messages are securely sent and received. In addition, to avoid problems, the server notifies the user if the device goes offline. SIM card replacement fraud and impersonation are limited because users' phone numbers are verified and linked to a unique device ID known only to the server.

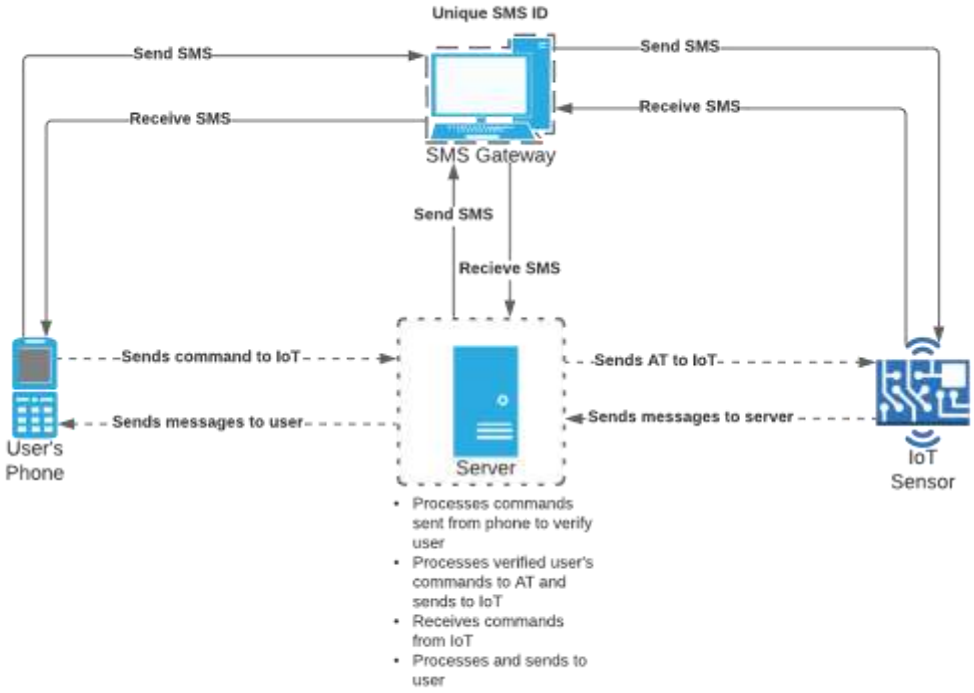

Figure 4.1: MAAMSIC Architecture

The users are connected to the device via the server through their verified mobile phones. They can join either via smartphones or the simple phones that do not have any access to the internet. For smartphones, the user uses a Flutter app whose data is encrypted with JSON Web Token (JWT) and JSON Web Encryption (JWE) to prevent third-party apps from reading user information. For simple phones, the user sends SMS to a specific SMS code, which concurrently sends commands to the server. The server then verifies the user and relays the command to the device. The server also sends feedback to the user from the device.

4.2 Server App

The server consists of a database module that stores data on user details and device details. It has a web app that comprises five main modules: Authentication, Message Processing Module, SMS Gateway, and IoT connector module.

The Authentication Module ensures that the user is registered and verified; their phone numbers that is being used to send SMS through commands are verified and are linked to the device(s) ID. When a user sends a message, the Authentication Module immediately checks whether the user is verified and connected to a device or devices. The Authentication Module sends a letter to the user in case there is a malicious attempt to send commands to their device. These attempts are also saved in a server with a list of these issues.

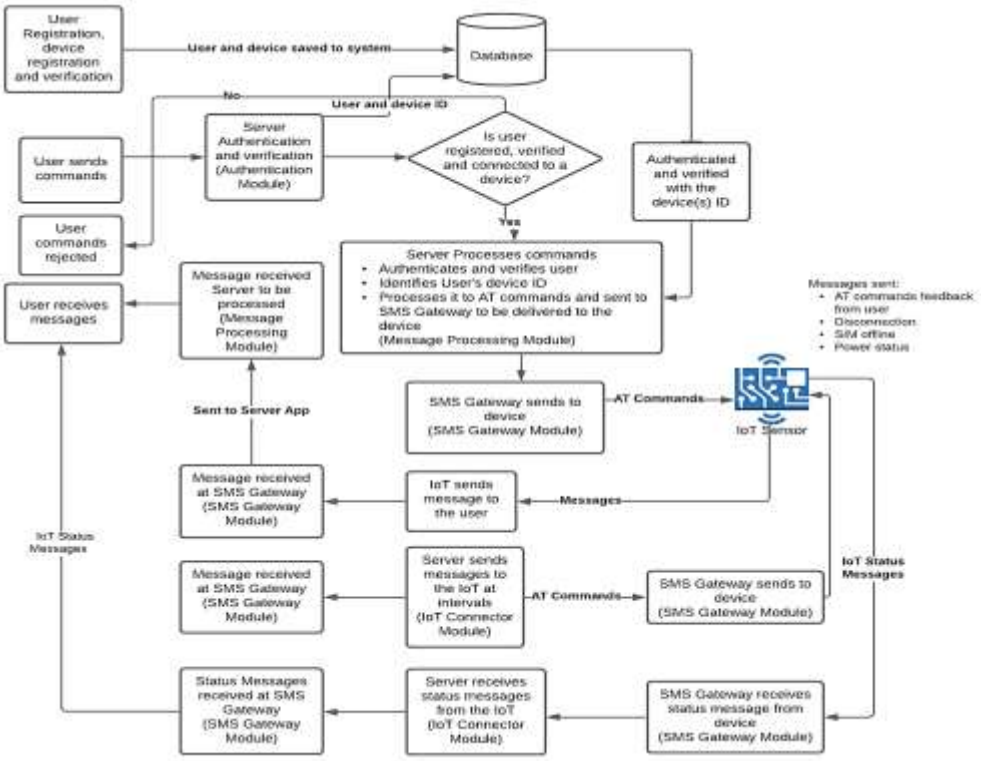

Figure.4.2: Software Architecture Block Diagram

The Message Processing Module handles received and sent messages. When a user sends a command to the device, it is verified through the auth server. After successful verification, their message is processed into an AT command and sent to the device through the SMS Gateway Module. When an IoT sends feedback or information to the user, the message is sent via SMS to the Message Processing Module through the SMS

Gateway. The message is processed and the user connected to the device identified. The Message Processing Module then sends a feedback SMS to the user through the SMS Gateway.

SMS Gateway Module is the one that sends and receives messages which are first processed in the server. When the user sends a message to the device, the SMS Gateway sends the message to the server for verification at the authentication module. After verification and processing, the SMS Gateway sends the message to the device's SIM card in the form of AT commands. The message from the device is first received by the SMS Gateway, and then sent to the Message Processing Module. The device sending the message is identified with the user connected to the device. The Message Processing Module then sends a feedback message to the user through the SMS Gateway.

The IoT Connector Module sends several messages to the IoT through the SMS Gateway to ensure that it is online and check its status. Messages are sent in intervals. A user can also check this to ensure that the device is online. The moment the IoT Connector module doesn't receive a message from the device, a message is sent to the user, and the issue logged in the system. All this data is stored in the database for querying and retrieval.

The server setup also needs the following crucial dependencies (latest as per the time the model was created):

- "express": "^4.17.1" - This is responsible for robust routing, high performance, high-test coverage, HTTP helpers (like redirection and caching) and content negotiation. It is responsible for generating applications quickly.

- "express-watcher": "^1.0.3" – This is a middle ware for monitoring response time, memory used and CPU. It is essential for time space complexity.

- "mongoose": "^5.6.0" – This is an Object Data Modeling (ODM) library for MomgoDB and Node JS. This is responsible for database management whereby it maps objects between Node and MongoDB.

- "multer": "^1.4.1" – It is a NodeJS middle ware responsible for handling form data. It is primarily used for uploading files.

- "express-oauth-server": "^2.0.0" – This provides middlewares for granting and authorizing tokens.

MongoDB Database Setup

The database of choice is MongoDB, which is popularly used for modern apps. It a document-based database that uses JSON-like documents. It is also a cross-platform database that can be set up on all operating systems.

The database model set up is as follows:

```
/*** Module dependencies. */
var mongoose = require('mongoose');
var Schema = mongoose.Schema;

/*** Schema definitions.*/
mongoose.model('OAuthTokens', new Schema({
  accessToken: { type: String },
  accessTokenExpiresOn: { type: Date },
  client : { type: Object },  // `client` and `user` are required in multiple places, for example `getAccessToken()`
  clientId: { type: String },
  refreshToken: { type: String },
  refreshTokenExpiresOn: { type: Date },
  user : { type: Object },
  userId: { type: String },
}));
```

```
mongoose.model('OAuthClients', new Schema({
  clientId: { type: String },
  clientSecret: { type: String },
  redirectUris: { type: Array }
}));

mongoose.model('OAuthUsers', new Schema({
  email: { type: String, default: '' },
  firstname: { type: String },
  lastname: { type: String },
  phone: { type: String },
  password: { type: String },
  username: { type: String }
}));

var OAuthTokensModel = mongoose.model('OAuthTokens');
var OAuthClientsModel = mongoose.model('OAuthClients');
var OAuthUsersModel = mongoose.model('OAuthUsers');

/*** Get access token.*/
module.exports.getAccessToken = function(bearerToken) {
  // Adding `.lean()`, as we get a mongoose wrapper object back from `findOne(...)`, and oauth2-server
complains.
  return OAuthTokensModel.findOne({ accessToken: bearerToken }).lean();
};
/*** Get client. */
module.exports.getClient = function(clientId, clientSecret) {
  return OAuthClientsModel.findOne({ clientId: clientId, clientSecret: clientSecret }).lean();
};

/** * Get refresh token. */
module.exports.getRefreshToken = function(refreshToken) {
  return OAuthTokensModel.findOne({ refreshToken: refreshToken }).lean();
};
```

```
/*** Get user.  */
module.exports.getUser = function(username, password) {
  return OAuthUsersModel.findOne({ username: username, password: password }).lean();
};
/** * Save token. */
module.exports.saveToken = function(token, client, user) {
  var accessToken = new OAuthTokensModel({
    accessToken: token.accessToken,
    accessTokenExpiresOn: token.accessTokenExpiresOn,
    client : client,
    clientId: client.clientId,
    refreshToken: token.refreshToken,
    refreshTokenExpiresOn: token.refreshTokenExpiresOn,
    user : user,
    userId: user._id,
  });
  // Can't just chain `lean()` to `save()` as we did with `findOne()` elsewhere. Instead we use `Promise` to resolve
the data.
  return new Promise( function(resolve,reject){
    accessToken.save(function(err,data){
      if( err ) reject( err );
      else resolve( data );
    }) ;
  }).then(function(saveResult){
    // `saveResult` is mongoose wrapper object, not doc itself. Calling `toJSON()` returns the doc.
    saveResult = saveResult && typeof saveResult == 'object' ? saveResult.toJSON() : saveResult;

    // Unsure what else points to `saveResult` in oauth2-server, making copy to be safe
    var data = new Object();
```

Figure 4.2: Multimodal Database Setup

Description of the Database Tables from the Above setup

1. **Users:**
    - Columns:
        - UserID (Primary Key)
        - Username
        - Password (Hashed)
        - PhoneNumber (Unique)
        - IsVerified (Boolean)

2. **Devices:**
    - Columns:
        - DeviceID (Primary Key)
        - UserID (Foreign Key references Users.UserID)
        - DeviceName
        - SIMCardNumber (Unique)
3. **Messages:**
    - Columns:
        - MessageID (Primary Key)
        - UserID (Foreign Key references Users.UserID)
        - DeviceID (Foreign Key references Devices.DeviceID)
        - Direction (Incoming/Outgoing)
        - Content
        - Timestamp
4. **SecurityEvents:**
    - Columns:
        - EventID (Primary Key)
        - UserID (Foreign Key references Users.UserID)
        - DeviceID (Foreign Key references Devices.DeviceID)
        - EventType (e.g., Unauthorized Access Attempt)
        - Timestamp
5. **DeviceStatus:**
    - Columns:
        - DeviceID (Foreign Key references Devices.DeviceID)
        - LastSeen (Timestamp)

Relationships:

- A User can have many Devices (One-to-Many).
- A Device belongs to one User (Many-to-One).

- A Message is sent by a User to a specific Device or received from a specific Device (Many-to-Many).
- A User can have many SecurityEvents (One-to-Many).
- A Device can have many SecurityEvents (One-to-Many).
- A Device has one DeviceStatus record (One-to-One).

The system has the following features:

- User passwords should be stored securely using hashing algorithms.
- This is a simplified representation and additional tables or fields may be required depending on specific functionalities.
- This table structure aims to represent the described functionalities in a relational database format. The actual implementation may vary depending on chosen database management system and specific needs of the application.

Adding JWT and JWS to the Model

The following library, to manage JWT and JWS is added to manage tokens

- "jsonwebtoken": "^8.3.0" – Responsible for writing and verifying JWT tokens.

Package.json

```
{
  "name": "jwt-oauth-integ",
  "version": "1.0.0",
  "description": "A model for Integrating Node JS apps with Oauth 2.0,  JWT & JWS",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "debug": "node --inspect-brk server.js"
  },
```

"repository": {

"type": "git",

"url": "git+https://github.com/narenaryan/node-jwt-integ.git"

},

"keywords": [

"node",

"jwt",

"oauth"

],

"author": "Joan Nabusoba",

"license": "ISC",

"bugs": {

"url": "https://github.com/narenaryan/node-jwt-integ/issues"

},

"homepage": "https://github.com/narenaryan/node-jwt-integ#readme",

"devDependencies": {

"eslint": "^5.4.0",

"eslint-config-standard": "^11.0.0",

"eslint-plugin-import": "^2.14.0",

"eslint-plugin-node": "^7.0.1",

"eslint-plugin-promise": "^4.0.0",

"eslint-plugin-standard": "^3.1.0"

},

"dependencies": {

"express": "^4.16.3",

"jsonwebtoken": "^8.3.0"

"express-watcher": "^1.0.3"

"mongoose": "^5.6.0"

"express-oauth-server": "^2.0.0"

}

}

After adding dependencies, you run *npm install* to ensure that all dependencies are added to the project. This also highlights the vulnerabilities in the project.

The model contains three main files in the set up: one that contains the main application logic(server), one that definesOAuth 2.0, JWT with JWS (a middleware) validation logic and a *config.js* file that is for storing secret for hashing the tokens.

The configuration file, *config.js*, contains the secret, written as follows:

```
module.exports = {
  secret: 'worldisfullofdevelopers'
};
```

This secret is responsible for creation and validation of tokens. The secret, during deployment, is stored in an environment variable instead of a file.

The middleware is responsible for requesting tokens and proceeding only when the token is validated. The middleware is written as follows:

```
let jwt = require('jsonwebtoken');
const config = require('./config.js');
let checkToken = (req, res, next) => {
  let token = req.headers['x-access-token'] || req.headers['authorization']; // Express headers are auto converted to lowercase
  if (token.startsWith('Bearer ')) {
    // Remove Bearer from string
    token = token.slice(7, token.length);
  }
  if (token) {
    jwt.verify(token, config.secret, (err, decoded) => {
      if (err) {
```

```
        return res.json({
          success: false,
          message: 'Token is not valid'
        });
      } else {
        req.decoded = decoded;
        next();
      }
    });
  } else {
    return res.json({
      success: false,
      message: 'Auth token is not supplied'
    });
  }
};

module.exports = {
  checkToken: checkToken
}
```

The above middleware does the following important things:

1. Captures headers containing access tokens and authorization
2. If the header has ‘Authorization: Bearer: xxx…’ it is formatted, and stripped off unwanted prefix before token.
3. Token is validated using **jwt** package and **secret** string
4. If there’s an error on anything goes wrong, it returns an error immediately before passing the control to another handler
5. Exports functions of the middleware for other modules to use.

Next is the token creation and handling.

Server for handling token generation

```
const express = require('express');
const bodyParser = require('body-parser');
let jwt = require('jsonwebtoken');
let config = require('./config');
let middleware = require('./middleware');

class HandlerGenerator {
  login (req, res) {
    let username = req.body.username;
    let password = req.body.password;
    // For the given username fetch user from DB
    let mockedUsername = 'admin';
    let mockedPassword = 'password';

    if (username && password) {
      if (username === mockedUsername && password === mockedPassword) {
        let token = jwt.sign({username: username},
          config.secret,
          { expiresIn: '24h' // expires in 24 hours
          }
        );
        // return the JWT token for the future API calls
        res.json({
          success: true,
          message: 'Authentication successful!',
          token: token
        });
      } else {
```

```
        res.send(403).json({
          success: false,
          message: 'Incorrect username or password'
        });
      }
    } else {
      res.send(400).json({
        success: false,
        message: 'Authentication failed! Please check the request'
      });
    }
  }
  index (req, res) {
    res.json({
      success: true,
      message: 'Index page'
    });
  }
}

// Starting point of the server
function main () {
  let app = express(); // Export app for other routes to use
  let handlers = new HandlerGenerator();
  const port = process.env.PORT || 8000;
  app.use(bodyParser.urlencoded({ // Middleware
    extended: true
  }));
  app.use(bodyParser.json());
  // Routes & Handlers
```

```
  app.post('/login', handlers.login);
  app.get('/', middleware.checkToken, handlers.index);
  app.listen(port, () => console.log(`Server is listening on port: ${port}`));
}
main();
```

4.3 Hardware

The soil moisture control system's hardware for irrigation consists of a GSM Arduino board, soil moisture sensor, and water level sensor. The board is programmed to send commands to the server and receive commands from the soil moisture sensor. The GSM chip is interfaced with the hardware. The chip is used to send and receive messages to control the system. The AT commands are sent and acquired through the GSM chip. A soil moisture sensor is used to detect the moisture content, which correspondingly sends SMS to the SMS gateway which is later on relayed to the user via the server. A water level sensor is used to indicate the amount of water left in the irrigation tank, and a 240V AC power supply drives the hardware.

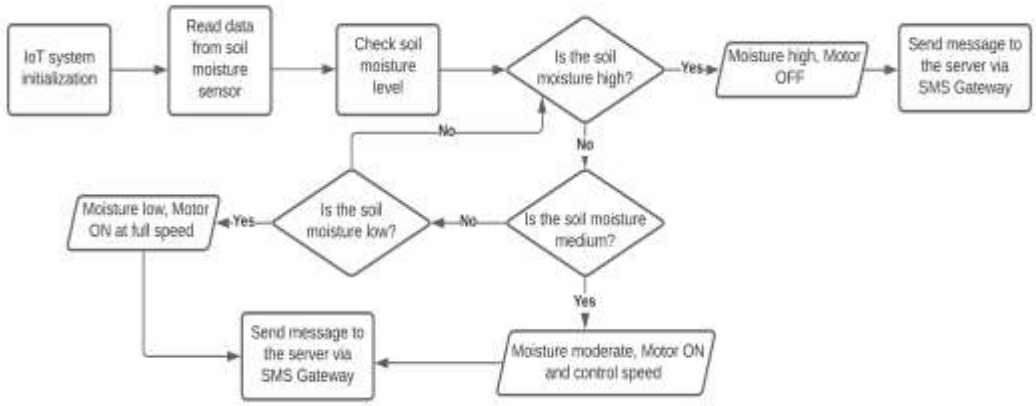

Figure 4.3: Hardware Architecture

4.4 Software Implementation

4.4.1 Server

The server is written in the HyperText Preprocessor (PHP) framework called Yii. It is hosted in a secured and shared hosting for demonstration. The server uses an SMS gateway provider to send and receive SMS through a short code. The server has a user registration module where the user is registered and authenticated to ensure data integrity. After the user is registered, the details are added to the server, including their device(s) SIM card number and ID. They further verify their phone number via a token sent to them through a short URL to activate their account. The IoT device has a GSM chip with SIM card, and this information is stored in the server. The admin then assigns the device to a verified user in the server, and the user can proceed to send commands and receive information from the device through the server.

When a server receives an SMS or a command from a specific phone number, the server knows the device ID associated with that user. The server then sends the message to the device. Moreover, when the device sends the message back to the server, the server identifies the user who owns the device and send it.

4.4.2 Mobile Application and Encryption

The user can install the mobile application to send and receive commands from the device through the server. This is for the case of those who own smartphones. The smartphone app is developed with Flutter and data is encrypted by JSON Web Token (JWT) and JSON Web Encryption (JWE). JWT and JWE technology is needed in the authentication process and access rights security (Royani & Wibowo, 2020). A user sends an encrypted message to the server, which decrypts it into a command that is sent to the IoT through the SMS gateway. Responses are sent back to the mobile application through the authentication and authorization module, which encrypts the message using JWE**. The mobile app, upon**

verification of the source of the message, decrypts the message and displays to the user. Data, which cannot be decrypted, is rejected by the receiving party.

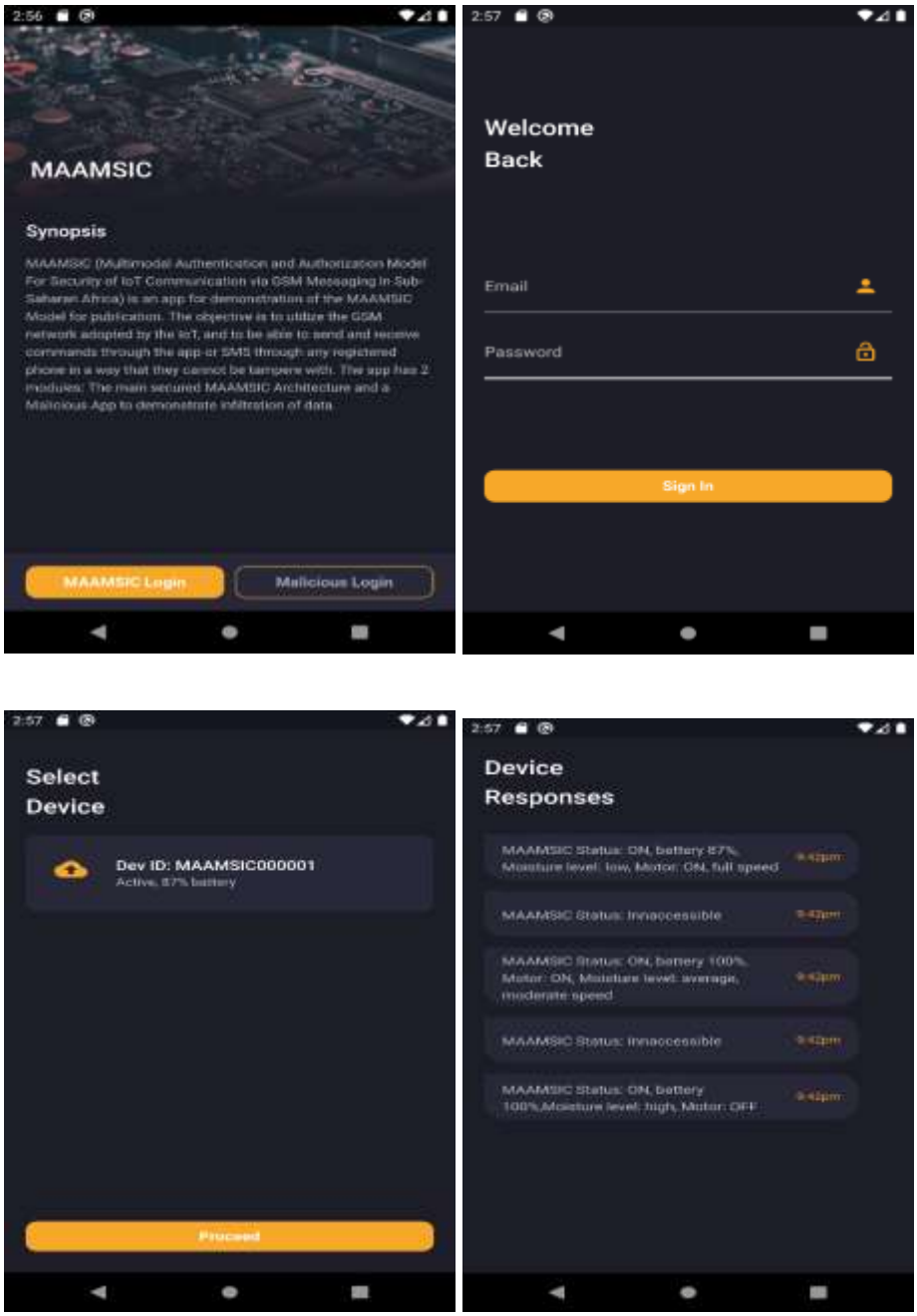

Figure 4.4: Mobile App Screenshots

4.5 Evaluation and Validation

MAAMSIC is evaluated by testing the mobile app and the IoT server. The mobile application has two login systems to emulate the MAAMSIC architecture and the phishing app. Both the modules have the SMS sent and received from the IoT. The user can send and receive SMS from the IoT through the server. The MAAMSIC application has a list of commands a user can directly send to the emulated IoT to control and operate it and control the moisture by turning the motor off or controlling the speed. The user can check the status of the IoT as well. Here is how the experimental setup is structured:

System Components:

- **IoT Devices**: These heterogeneous devices represent various aspects of IoT, such as soil humidity and irrigation control systems.
- **Secure Server**: The central component responsible for handling data encryption and communication with IoT devices.
- **User Interfaces**:
    - **Mobile App (Smartphones)**: Used for sending and receiving messages from the server.
    - **Basic Phones**: For users without internet access.

User Devices:

Table 4.1: Test Users and Devices

User	Phone number	Phone type	Phone model	User Status
Unregistered User 1	0718xxxx19	Basic Phone	Nokia 105 Africa Edition (Dual SIM)	UNREGISTERED
Unregistered User 2	0715xxxx62	Basic Phone	Nokia 105 Africa Edition (Dual SIM)	UNREGISTERED
Registered User 1	0724xxxx18	Smart Phone	Samsung Galaxy A03	REGISTERED
Registered User 2	0721xxxx87	Smart Phone	Iphone 12s	REGISTERED
Registered User 3	0123xxxx34	Smart Phone	Samsung Galaxy A10S	REGISTERED

Evaluation Metrics:

The MAAMSIC model is evaluated based on several key metrics:

- **Security and Authorization Efficiency**: Assessing the effectiveness of encryption mechanisms in protecting data during transmission.
- **Latency**: Evaluating the time taken for communication between devices and the secure server.
- **Resource Utilization**: Analysing memory and processing overhead introduced by MAAMSIC.

Table 4.2: Test Results for Security Metric

Request	Number of Requests	User	Result
Get IoT Status (ON/OFF)	50	Unregistered User 1	- 50 Commands FAILED. - IoT status UNDEFINED - Cause of Failure Authorization Failure
	50	Unregistered User 2	- 50 Commands FAILED. - IoT status UNDEFINED - Cause of Failure Authorization Failure
	50	Registered User 1	- 50 Commands Successful. - IoT status OFF
	50	Registered phone 2	- 44 Commands Successful. - IoT status OFF - 6 Commands FAILED. - Cause of Failure : Network Error
	50	Registered phone 3	- 50 Commands Successful. - IoT status OFF
Turn on motor	25	Unregistered User 1	- 25 Commands FAILED. - Motor status OFF - Cause of Failure Authorization Failure
	25	Unregistered User 2	- 25 Commands FAILED. - Motor status OFF - Cause of Failure Authorization Failure
	25	Registered User 1	- 25 Commands Successful. - Motor Turned ON
	25	Registered phone 2	- 25 Commands Successful. - Motor Turned ON
	25	Registered phone 3	- 25 Commands Successful. - Motor Turned ON
Turn off motor	25	Unregistered User 1	- 25 Commands FAILED. - Motor status ON - Cause of Failure Authorization Failure
	25	Unregistered User 2	- 25 Commands FAILED. - Motor status ON - Cause of Failure Authorization Failure
	25	Registered User 1	- 25 Commands Successful. - Motor Turned OFF
	25	Registered phone 2	- 25 Commands Successful. - Motor Turned OFF
	25	Registered phone 3	- 25 Commands Successful. - Motor Turned OFF

Table 4.3: Test Results for Latency Metric

Request	Number of Requests	User	Average Time Per Request	Result
Get IoT Status (ON/OFF)	50	Unregistered User 1	7.89	- 50 Commands FAILED. - IoT status UNDEFINED - Cause of Failure Authorization Failure
	50	Unregistered User 2	7.76	- 50 Commands FAILED. - IoT status UNDEFINED - Cause of Failure Authorization Failure
	50	Registered User 1	10.2	- 50 Commands Successful. - IoT status OFF
	50	Registered phone 2	10.5	- 44 Commands Successful. - IoT status OFF - 6 Commands FAILED. - Cause of Failure: Network Error
	50	Registered phone 3	10.6	- 50 Commands Successful. - IoT status OFF
Turn on motor	25	Unregistered User 1	8.4	- 25 Commands FAILED. - Motor status OFF - Cause of Failure Authorization Failure
	25	Unregistered User 2	8.4	- 25 Commands FAILED. - Motor status OFF - Cause of Failure Authorization Failure
	25	Registered User 1	8.6	- 25 Commands Successful. - Motor Turned ON
	25	Registered phone 2	8.6	- 25 Commands Successful. - Motor Turned ON
	25	Registered phone 3	8.7	- 25 Commands Successful. - Motor Turned ON
Turn off motor	25	Unregistered User 1	8.4	- 25 Commands FAILED. - Motor status ON - Cause of Failure Authorization Failure

Request	Number of Requests	User	Average Time Per Request	Result
	25	Unregistered User 2	8.5	- 25 Commands FAILED. - Motor status ON - Cause of Failure Authorization Failure
	25	Registered User 1	8.6	- 25 Commands Successful. - Motor Turned OFF
	25	Registered phone 2	8.6	- 25 Commands Successful. - Motor Turned OFF
	25	Registered phone 3	8.6	- 25 Commands Successful. - Motor Turned OFF

Comparison or Requests from Data of previous models

Table 4.4: Comparison with Other Models

		Waghmare Model	Antonio Model	Priyanka Model	MAAMSIC
Sensors used		Gas sensor, voltage regulator	Soil moisture sensor, pH sensor	Pressure sensor, temperature sensor, viscosity	Soil moisture sensor, water level sensor
SMS Time to get IoT Status (ms)	Registered phone 1	4.43	4.74	5.01	7.63
	Unregistered phone 1	4.87	4.74	5.01	7.89
	Unregistered phone 2	4.87	4.74	5.01	7.89
	Unregistered phone 3	4.87	4.74	5.01	7.89
Security implementations		(Not mentioned if any) Commands from unregistered devices went through	(Not mentioned if any) Commands from unregistered devices went through	(Not mentioned if any) Commands from unregistered devices went through	OAuth2.0, JWT & JWS, 2 factor auth

User's requests are issued on both registered and unregistered phone numbers to access the IoT device by sending messages. The messages sent are received by the server app which consists of authentication, message processing, SMS gateway and IoT connector modules. If a user is using a simple phone without GPRS, the phone number is processed to check whether the user is registered and verified, and is connected to an IoT device. If

the user's phone number is unverified, the command sent through the SMS is rejected. In the case of a smartphone, the mobile app is encrypted by JWT and JWE to ensure that messages sent and received aren't read by third-party apps. In both cases if an IoT device is attempted to be accessed by an unregistered phone number, a message is sent to the registered phone number for further actions. A flag is also raised on the server to block requests from the unregistered phone number.

Messages received by registered and verified numbers are processed by the server system to be commands and sent to IoT, which then returns feedback messages to the user.

MAAMSIC may produce large amounts of data and receive many requests and responses, making it require a large processing capacity surpassing the capacities available for this implementation.

4.6 Discussion

The MAAMSIC model tries to address the security issues of IoT in sub-Saharan Africa, as well as create a dynamic model that can be accessed by both simple phones and smart phones. This is to factor in the security related problems during IoT development, which are often underestimated or overlooked (Selgert, 2020). Even with inadequate coverage of the internet in rural parts of Africa, smart farming innovations are still being implemented using IoT and wireless devices (Olivera-Jr et al., 2020).

From the new model, user's requests are issued on both registered and unregistered phone numbers to access the IoT device by sending messages to the IoT system for validation. The messages sent are received by the server app which consists of authentication, message processing, SMS gateway and IoT connector modules.

There is also the issue of low network coverage, half of Sub-Saharan Africa does not have access to electricity, which consequently hampers GPRS equipment (Bakibinga-Gaswaga et al., 2020). The latter highlights how unstable GPRS is as a choice of IoT infrastructure, affecting its adoption.

Additionally, according to M-Kopa (https://m-kopa.com/impact/), one of the leading solar home system states that 75% of sub-Saharan Africa remains unconnected to the internet. Sending IoT data can use general packet radio service (GPRS), which is an advancement of GSM. However, since GPRS is not dedicated for transmission of IoT data due to weaknesses in power efficiency and coverage, GSM is preferred (Bima, Suryani & Wardana, 2020).

Putting this into consideration, if a user is using a simple phone without GPRS, the phone number is processed to check whether the user is registered and verified, and is connected to an IoT device. If the user's phone number is unverified, the command sent through the SMS is rejected. In the case of a smartphone, the mobile app is encrypted by JWT and JWE to ensure that messages sent and received aren't read by third-party apps. In both cases if an IoT device is attempted to be accessed by an unregistered phone number, a message is sent to the registered phone number for further actions. A flag is also raised on the server to block requests from the unregistered phone number.

MAAMSIC model's data of both the server and the mobile app is encrypted by JSON Web Token (JWT) and JSON Web Encryption (JWE). JWT and JWE technology is needed in the authentication process and access rights security (Royani & Wibowo, 2020).

4.7 Conclusion

As shown in the validation, we propose a Multimodal Authentication and Authorization Model for the Security of IoT Communication via GSM Messaging (MAAMSIC). This model uses GSM considering the challenges faced by internet coverage in major parts of the sub-Saharan Africa. Additionally, the security related issues faced by IoT systems in the sub-Saharan Africa is addressed by using a server and a mobile app that is encrypted, creating a secure channel for users to communicate with IoT systems without fear of data breaches. The capability of the new model to factor in security mitigations, like users being notified to contact admin incase their IoT is compromised, making registration and

verification process secure, among other logs, makes users involved in ensuring they have a secure, trusted and reliable to the user.

CHAPTER FIVE

RECOMMENDATIONS AND FURTHER WORK

5.1 Summary

In this thesis, we have managed to improve the security of web resources using a Multimodal model based on previous models. Previous models studied in this work show weaknesses in the existing IoT systems, highlighting the gap to guide this research and to provide input in knowledge contribution. The use of GSM messaging for direct IoT communication is not secure. This thesis makes significant contributions to the field of Internet of Things (IoT) security, particularly in the context of GSM messaging. Below, we outline the methodological and theoretical advancements presented in this work:

Methodological Contributions:

- Secure Server Application: We introduce a secure server application that acts as an intermediary between the User and the IoT device. By providing an abstraction layer, this gateway enhances security by isolating the two entities. Additionally, it facilitates rapid response to security breaches on either side.
- Encryption Mechanism: The data transmitted via GSM messaging to the IoT device is encrypted. This ensures that sensitive information remains confidential, accessible only to authorized senders and receivers.

Theoretical Contributions:

- Identifying Security Gaps: Our research identifies prevalent security gaps in existing implementations of IoT systems that rely on GSM messaging, particularly within Sub-Saharan Africa. These vulnerabilities pose risks to data integrity, privacy, and overall system reliability.

- Multi-Faceted Solution: We propose an innovative solution that addresses these security challenges. Our approach is characterized by its cost-effectiveness, lightweight design, and robust security features.

In summary, this thesis not only sheds light on critical security issues but also offers practical solutions for securing IoT deployments in resource-constrained regions.

From the research, it was concluded that putting into consideration the issues highlighted in the existing systems can improve the acquisition of IoT systems in the sub-Saharan Africa for the increase in food production. The research highlights issues of internet coverage, power in the rural areas and security of IoT. These considerations have been noted and added as solutions to the new model. These additions are aimed at utilizing the available resources efficiently while creating an adaptive model that can easily be adopted by farmers to increase their yields, without the fear of security, unavailability or data breach

An experiment was created based on the previous models, as well as the Multimodal model based on the gaps highlighted in the previous models. Experiments try to access unauthorized data in the system, and tries to operate the IoT using unverified accounts. This is done for both smartphone users and simple phone users, to assess the server strength in both scenarios.

Once the experiment was conducted on the previous models, a Multimodal model was proposed and created. The tools used were PHP (Yii framework), Flutter, API to interface the app and the server, SMS Gateway, Iotify (Realtime simulation of IoT). These implementations were aimed to modularize the work as well as creating a user-oriented framework. The User interfaces, back end and source codes are attached in this research.

To assess and validate the model, there were experiments conducted to try access the IoT data and operate it without verification. A phishing app that demonstrated the current implementation of IoT was tested against the MAAMSIC model.

From the validation as proposed in this research, the MAAMSIC model improved the security of IoT devices while utilizing the available technology. Additionally, the model implemented mitigation measures with the aim of involving the user in the security implementation. The system is also user-centric and users can easily interact with the IoT device securely.

5.2 Knowledge Contribution

From the experiment conducted, the key knowledge contributions are as follows:

1. The thesis developed a model to address the internet coverage issue in sub-Saharan Africa, by utilizing the GSM technology efficiently with the aim of increasing the use of IoT in the rural sub-Saharan Africa.
2. The thesis presented a Multimodal Authentication and Authorization Model for the Security of IoT Communication via GSM Messaging (MAAMSIC), which secures the server and the mobile app, and added mitigation measures.
3. The thesis creates a model that is user-oriented and dynamic, with the capability of using clean energy like solar in IoT devices, putting in consideration the poer coverage issues in the sub-Saharan Africa.
4. The thesis presents a model that conceptualizes the model proposed in this research.

5.3 Further Work

MAAMSIC architecture will be extended to support functionalities for dealing with more complex events and billing policies. Incorporating features that will enable it to scale and deal with the increased workload is essential for future work. A possible solution is deploying the MAAMSIC server in a dedicated cloud server with a strong RAM and large memory. MAAMSIC can also be transformed into multi-edged cloud architecture to deal with internal risks. The security of the IoT network to ensure that it handles risks due to malicious behavior of IoT is still an open issue. For further enhancement of this work, the

model could be designed to utilize solar or wind energy as its source of power. This is a cleaner energy and will be a great source of power in rural areas where electricity is not reliable. The MAAMSIC could also be improved by use of Artificial Intelligence and audio processing for better accessibility for the disabled especially those with poor eyesight.

REFERENCES

Abbott, P., Checco, A., & Polese, D. (2021). Smart Farming in sub-Saharan Africa: Challenges and Opportunities.

Adenugba, F., Misra, S., Maskeliūnas, R., Damaševičius, R., & Kazanavičius, E. (2019). Smart irrigation system for environmental sustainability in Africa: An Internet of Everything (IoE) approach. Mathematical biosciences and engineering, 16(5), 5490-5503.

African Wildlife Foundation. (2023). African Wildlife Foundation's Vision for Conservation. Retrieved from https://www.awf.org/strategic-vision-2020-2030

Aguera, P., Berglund, N., Chinembiri, T., Comninos, A., Gillwald, A., & Govan-Vassen, N. (2020). Paving the way towards digitalising agriculture in South Africa.

Akhigbe, B. I., Munir, K., Akinade, O., Akanbi, L., & Oyedele, L. O. (2021). IoT Technologies for Livestock Management: A Review of Present Status, Opportunities, and Future Trends. *Big Data and Cognitive Computing*, *5*(1), 10.

Alam, T. (2020). A middleware framework between mobility and IoT using IEEE 802.15. 4e Sensor Networks. *Jurnal Online Informatika*, *4*(2), 90-94.

Almazrouei, O. S. M. B. H., Magalingam, P., Hasan, M. K., & Shanmugam, M. (2023). A Review on Attack Graph Analysis for IoT Vulnerability Assessment: Challenges, Open Issues, and Future Directions. *IEEE Access*

Ammour, N., Bazi, Y., & Alajlan, N. (2023). Multimodal Approach for Enhancing Biometric Authentication. Journal of Imaging, 9(9), 168.

Antony, A. P., Leith, K., Jolley, C., Lu, J., & Sweeney, D. J. (2020). A review of practice and implementation of the Internet of Things (IoT) for smallholder agriculture. *Sustainability*, *12*(9), 3750.

Asare, D. (2023). Connecting people in Agriculture. Retrieved from https://esoko.com/who-we-are/

Backman, J., Linkolehto, R., Koistinen, M., Nikander, J., Ronkainen, A., Kaivosoja, J., ... & Pesonen, L. (2019). Cropinfra research data collection platform for ISO 11783 compatible and retrofit farm equipment. Computers and Electronics in Agriculture, 166, 105008.

Bakibinga-Gaswaga, E., Bakibinga, S., Bakibinga, D. B. M., & Bakibinga, P. (2020). Digital technologies in the COVID-19 responses in sub-Saharan Africa: policies, problems, and promises. *The Pan African Medical Journal*, *35*(38).

Bassene, A., & Gueye, B. (2021). A Group-Based IoT Devices Classification through Network Traffic Analysis Based on Machine Learning Approach. In *Towards new e-Infrastructure and e-Services for Developing Countries: 12th EAI International Conference, AFRICOMM 2020, Ebène City, Mauritius, December 2-4, 2020, Proceedings 12* (pp. 185-202). Springer International Publishing.

Bhagat, M., Kumar, D., & Kumar, D. (2019, March). Role of Internet of Things (IoT) in smart farming: a brief survey. In *2019 Devices for Integrated Circuit (DevIC)* (pp. 141-145). IEEE.

Bima, I. W. K., Suryani, V., & Wardana, A. A. (2020). A Performance Analysis of General Packet Radio Service (GPRS) and Narrowband Internet of Things (NB-IoT) in Indonesia. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, *5*(1), 11-20.

Braghin, C., Lilli, M., & Riccobene, E. (2023). A model-based approach for vulnerability analysis of IoT security protocols: The Z-Wave case study. *Computers & Security*, *127*, 103037.

Chen, J., & Yang, A. (2019). Intelligent agriculture and its key technologies based on internet of things architecture. *IEEE Access*, *7*, 77134-77141.

Danielle, N. E. L., & Masilela, L. (2020). Open Governance For Improved Service Delivery Innovation In South Africa. *International Journal of Ebusiness and Egovernment Studies*, *12*(1), 33-47.

De Diego, S., Regueiro, C., & Macia-Fernandez, G. (2021). Enabling identity for the iot-as-a-service business model. *IEEE Access*, *9*, 159965-159975

du Preez, M. L. (2020). 4IR and Water Smart Agriculture in Southern Africa: A Watch List of Key Technological Advances.

Eduam Emanuel, (2023). Smart City in Accra-Ghana: A Gold Coast again. TheSmartCityJournal. Retrieved from https://www.jica.go.jp/english/ information/press/2023/20231010_42.html

Ewan, K. (2023). An agritech that has developed a peer-to-peer network platform for smallholder farmers in Eastern Africa. Retrieved from https://agfunder.com/portfolio/wefarm/

Farooq, M. S., Riaz, S., Abid, A., Umer, T., & Zikria, Y. B. (2020). Role of IoT technology in agriculture: A systematic literature review. *Electronics*, *9*(2), 319.

Fastellini, G., & Schillaci, C. (2020). Precision farming and IoT case studies across the world. In *Agricultural Internet of Things and Decision Support for Precision Smart Farming* (pp. 331-415). Academic Press.

Ferrag, M. A., Maglaras, L. A., Janicke, H., Jiang, J., & Shu, L. (2017). Authentication protocols for internet of things: a comprehensive survey. *Security and Communication Networks*, 2017.

Ferrag, M. A., Maglaras, L., & Derhab, A. (2019). Authentication and authorization for mobile IoT devices using biofeatures: Recent advances and future trends. *Security and Communication Networks*, *2019*.

Ferrag, M. A., Shu, L., Yang, X., Derhab, A., & Maglaras, L. (2020). Security and privacy for green IoT-based agriculture: Review, blockchain solutions, and challenges. *IEEE access*, *8*, 32031-32053.

Fusi, G., & Mbarika, V. (2022). A Review of IoT Trends and Usage in Developing Economies: The Case of Sub-Saharan Africa.

Grados-Licham, B., & Bedón-Monzón, H. M. (2020). Software Components of an IoT Monitoring Platform in Google Cloud Platform: A Descriptive Research and an Architectural Proposal.

GSMA. (2021, July 13). The Mobile Economy Sub-Saharan Africa 2021. Retrieved from https://www.gsma.com/mobileeconomy/sub-saharan-africa/

GSMA. (2023). The Mobile Economy Sub-Saharan Africa 2023. https://event-assets.gsma.com/pdf/20231017-GSMA-Mobile-Economy-Sub-Saharan-Africa-report.pdf

Guo, L., Zhang, C., Sun, J., & Fang, Y. (2013). A privacy-preserving attribute-based authentication system for mobile health networks. IEEE Transactions on Mobile Computing, 13(9), 1927-1941.

Guo, Y., & Guo, Y. (2023). CS-LAKA: A lightweight authenticated key agreement protocol with critical security properties for IoT environments. IEEE Transactions on Services Computing.

Gupta, G., Setia, R., Meena, A., & Jaint, B. (2020, June). Environment Monitoring System for Agricultural Application using IoT and Predicting Crop Yield using Various

Data Mining Techniques. In *2020 5th International Conference on Communication and Electronics Systems (ICCES)* (pp. 1019-1025). IEEE.

Holst, C., Sukums, F., Radovanovic, D., Ngowi, B., Noll, J., & Winkler, A. S. (2020). Sub-Saharan Africa—the new breeding ground for global digital health. *The Lancet Digital Health*, *2*(4), e160-e162.

Hove-Sibanda, P., Matshidiso, M., & Igwe, P. A. (2021). Supply chain risks, technological and digital challenges facing grocery retailers in South Africa. *Journal of Enterprising Communities: People and Places in the Global Economy*.

Ingram, W., & Memon, F. A. (2020). Robustness of IoT-connected e-Taps for sustainable service delivery of rural water supply. *Water Supply*.

Ingram, W., & Memon, F. A. (2020). Robustness of IoT-connected e-Taps for sustainable service delivery of rural water supply. *Water Supply*, *20*(6), 2251-2260.

Ingram, W., & Memon, F. A. (2020). Robustness of IoT-connected e-Taps for sustainable service delivery of rural water supply. *Water Supply*, *20*(6), 2251-2260.

International Labour Organization. (2020, July 29). Digital skills and the future of work: Challenges and opportunities in a post COVID-19 environment. Retrieved from https://www.ilo.org/wcmsp5/groups/public/---ed_emp/documents/publication/wcms_766085.pdf

Iorliam, A., Iorliam, I. B., & Bum, S. Internet of Things for Smart Agriculture in Nigeria and Africa: A Review.

JICA (Japan International Cooperation Agency), (10 October 2023). Signing of Grant Agreement with Rwanda: Realizing smooth and stable traffic in Kigali City through the introduction of an Intelligent Transport System and improvement of

road intersections. Retrieved from https://www.jica.go.jp/english/information/press/2023/20231010_42.html

Karim, F., & Karim, F. (2017). Monitoring system using web of things in precision agriculture. *Procedia Computer Science*, *110*, 402-409.

Kitenge, S. Y. (2020). Bridging the Digital Divide with upskilling strategies which unlock an expert IoT workforce: A way forward for AUDA-NEPAD.

Kite-Powell, J. Why Precision Agriculture Will Change How Food Is Produced. Retrieved from https://www.forbes.com/sites/jenniferhicks/2018/04/30/why-precision-agriculture - will-change-how-food-is-produced/#1aa438ec6c65

Kuteyi, D., & Winkler, H. (2022). Logistics challenges in sub-Saharan Africa and opportunities for digitalization. *Sustainability*, *14*(4), 2399.International Telecommunication Union (ITU). (2020, September 29). Global

Lee, S., Jeong, Y., Son, S., & Lee, B. (2019). A self-predictable crop yield platform (SCYP) based on crop diseases using deep learning. Sustainability, *11*(13), 3637.

Li, S., Yuan, F., Ata-UI-Karim, S. T., Zheng, H., Cheng, T., Liu, X., ... & Cao, Q. (2019). Combining color indices and textures of UAV-based digital imagery for rice LAI estimation. *Remote Sensing*, *11*(15), 1763.

Light, R, (2021, April 21). *"MQTT"* A lightweight publish/subscribe messaging protocol useful for use with low power sensors, but is applicable to many scenarios. Mosquitto. Retrieved from https://mosquitto.org/man/mqtt-7.html

Liu, N., Cao, W., Zhu, Y., Zhang, J., Pang, F., & Ni, J. (2016). Node deployment with k-connectivity in sensor networks for crop information full coverage monitoring. *Sensors*, *16*(12), 2096.

Lukitowati, R., & Ramli, K. (2020). Assessing the Information Security Awareness of Employees in PT ABC against International Organization for Standardization (ISO) 27001: 2013. *Journal of Computational and Theoretical Nanoscience, 17*(2-3), 1441-1446.

Madakam, S., Lake, V., Lake, V., & Lake, V. (2015). Internet of Things (IoT): A literature review. *Journal of Computer and Communications, 3*(05), 164.

Maiga, J., Suyoto, S., & Pranowo, P. (2021, March). Mobile app design for sustainable agriculture in Mali-West Africa. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1098, No. 3, p. 032037). IOP Publishing.

Mekki, K., Bajic, E., Chaxel, F., & Meyer, F. (2019). A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT express*, *5*(1), 1-7.

Mkomwa, S., Kassam, A., Bwalya, M., & Shula, R. K. (2022). The Malabo Declaration and Agenda 2063: Making Climate Smart Agriculture Real with Conservation Agriculture in Africa. In Conservation Agriculture in Africa: Climate Smart Agricultural Development (pp. 1-16). GB: CABI.

Navarro, E., Costa, N., & Pereira, A. (2020). A systematic review of IoT solutions for smart farming. *Sensors*, *20*(15), 4231.

Navulur, S., & Prasad, M. G. (2017). Agricultural management through wireless sensors and internet of things. *International Journal of Electrical and Computer Engineering*, *7*(6), 3492.

Nazareno, A. C., da Silva, I. J., Nunes, E. F., Gogliano Sobrinho, O., Marè, R. M., & Cugnasca, C. E. (2020). Real-time web-based microclimate monitoring of broiler chicken trucks on different shifts. *Revista Brasileira de Engenharia Agrícola e Ambiental*, *24*(8), 554-559.

Ndubueze, P. N. (2020). Cybercrime and Legislation in an African Context. *The Palgrave Handbook of International Cybercrime and Cyberdeviance*, 345-364.

Ndung'u, N., & Signé, L. (2020). The Fourth Industrial Revolution and digitization will transform Africa into a global powerhouse. *Foresight Africa Report*.

Nigussie, E., Olwal, T. O., Lemma, A., Mekuria, F., & Peterson, B. (2020). IoT Architecture for Enhancing Rural Societal Services in Sub-Saharan Africa. *Procedia Computer Science, 177*, 338-344.

Nigussie, E., Olwal, T., Musumba, G., Tegegne, T., Lemma, A., & Mekuria, F. (2020). IoT-based irrigation management for smallholder farmers in rural sub-Saharan Africa. *Procedia Computer Science*, *177*, 86-93.

Oliveira-Jr, A., Resende, C., Pereira, A., Madureira, P., Gonçalves, J., Moutinho, R., ... & Moreira, W. (2020). IoT Sensing Platform as a Driver for Digital Farming in Rural Africa. *Sensors*, *20*(12), 3511.

Oliveira-Jr, A., Resende, C., Pereira, A., Madureira, P., Gonçalves, J., Moutinho, R., ... & Moreira, W. (2021). IoT Sensing Box to Support Small-Scale Farming in Africa. In *Towards new e-Infrastructure and e-Services for Developing Countries: 12th EAI International Conference, AFRICOMM 2020, Ebène City, Mauritius, December 2-4, 2020, Proceedings 12* (pp. 171-184). Springer International Publishing.

Oliveira-Jr, A., Resende, C., Pereira, A., Madureira, P., Gonçalves, J., Moutinho, R., ... & Moreira, W. (2021). IoT Sensing Box to Support Small-Scale Farming in Africa. In *Towards new e-Infrastructure and e-Services for Developing Countries: 12th EAI International Conference, AFRICOMM 2020, Ebène City, Mauritius, December 2-4, 2020, Proceedings 12* (pp. 171-184). Springer International Publishing.

Owuor, D. O., Laurent, A., & Orero, J. O. (2020, August). Exploiting IoT data crossings for gradual pattern mining through parallel processing. In *ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium* (pp. 110-121). Springer, Cham.

Park, M., Kayuni, M., Manda, T., & Kim, H. (2020). Modeling Secure Home Area Network Based on IoT for Resource Constraints Environment. *Journal of Computer and Communications*, *8*(01), 45.

Parsa, A. (2023). By combining the ever-growing computing power of machines with the best medical expertise of humans to create a comprehensive, immediate and personalised health service and making it universally available. Retrieved from https://www.babyl.rw/about/

Pattnaik, P. K., Kumar, R., Pal, S., & Panda, S. N. (Eds.). (2020). *IoT and analytics for agriculture*. Springer Singapore.

Paul Antony, A., Leith, K., Jolley, C., Lu, J., & Sweeney, D. (2020). A Review of Practice and Implementation of the Internet of Things (IoT) for Smallholder Agriculture.

Priyanka, E. B., Thangavel, S., Madhuvishal, V., Tharun, S., Raagul, K. V., & Krishnan, C. S. (2020). Application of Integrated IoT Framework to Water Pipeline Transportation System in Smart Cities. In *Intelligence in Big Data Technologies—Beyond the Hype* (pp. 571-579). Springer, Singapore.

Rak, M., Salzillo, G., & Romeo, C. (2020, February). Systematic IoT Penetration Testing: Alexa Case Study. In *ITASEC* (pp. 190-200).

Ramakrishnan, M. (2023). Signature Based V2X Communication and Authentications Using Resourceful Signcryption and Optimised Ecc.

Rivas-Sánchez, Y. A., Moreno-Pérez, M. F., & Roldán-Cañas, J. (2019). Environment control with low-cost microcontrollers and microprocessors: Application for green walls. *Sustainability*, *11*(3), 782.

Robinson, D. (2020). *NB-IoT (LTE Cat-NB1/narrow-band IoT) performance evaluation of variability in multiple LTE vendors, UE devices and MNOs* (Doctoral dissertation).

Routray, S. K., Javali, A., Ghosh, A. D., & Sarangi, S. (2020, July). An Outlook of Narrowband IoT for Industry 4.0. In *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)* (pp. 923-926). IEEE.

Royani, M. R., & Wibowo, A. (2020). Web Service Implementation in Logistics Company uses JSON Web Token and RC4 Cryptography Algorithm. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *4*(3), 591-600.

Salamatian, S., Huleihel, W., Beirami, A., Cohen, A., & Médard, M. (2019). Why botnets work: Distributed brute-force attacks need no synchronization. *IEEE Transactions on Information Forensics and Security*, *14*(9), 2288-2299.

Sanchez-Iborra, R., & Cano, M. D. (2016). State of the art in LP-WAN solutions for industrial IoT services. *Sensors*, *16*(5), 708.

Selgert, F. (2020). Cynefin Framework, DevOps and Secure IoT: Understanding the Nature of IoT Systems and Exploring Where in the DevOps Cycle Easy Gains Can Be Made to Increase Their Security. In *Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops: DECSoS 2020, DepDevOps 2020, USDAI 2020, and WAISE 2020, Lisbon, Portugal, September 15, 2020, Proceedings 39* (pp. 255-265). Springer International Publishing.

Shelby, Z.; Hartke, K.; Bormann, C. The Constrained Application Protocol (CoAP). Retrieved from https://www.rfc-editor.org/info/rfc7252

Shelke, N. A., Ahuja, S., Banarjee, A., Singh, N. P., Kasana, S. S., & Kukreja, S. (2022, May). AI-enabled IoT based multimodal authentication system for securing the hardware and software clients. In *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)* (Vol. 1, pp. 545-550).

Sim, M., Eum, S., Song, G., Yang, Y., Kim, W., & Seo, H. (2023). K-XMSS and K-SPHINCS+: Enhancing Security in Next-Generation Mobile Communication and Internet Systems with Hash Based Signatures Using Korean Cryptography Algorithms. *Sensors*, *23*(17), 7558.

Singla, D., & Verma, N. (2023). Performance Analysis of Authentication system: A Systematic Literature Review.

Siwakoti, Y. R., Bhurtel, M., Rawat, D. B., Oest, A., & Johnson, R. C. (2023). Advances in IOT security: Vulnerabilities, enabled Criminal Services, attacks and countermeasures. *IEEE Internet of Things Journal*.

Skouloiudi, C., Malatras, A., Naydenov, R., & Dede, G. (2020). Guidelines For Securing the Internet of Things: Secure supply chain for IoT. Retrieved from https://www.enisa.europa.eu/publications/guidelines-for-securing-the-internet-of-things/

Stočes, M., Vaněk, J., Masner, J., & Pavlík, J. (2016). Internet of things (iot) in agriculture-selected aspects. *Agris on-line Papers in Economics and Informatics*, *8*(665-2016-45107), 83-88.

Stojkovic, N., Orlic, V., Peric, M., Drajic, D., & Rakic, A. (2020). Concept of System for Surveillance and Monitoring of IoT HFSWR Network. *Proc. of IcETRAN*.

The Ocean Cleanup. (2023, June 6). From Connectivity to Services: Digital Transformation in Africa. Retrieved from https://www.worldbank.org/en/results/2023/06/26/from-connectivity-to-services-digital-transformation-in-africa

Thorat, A., Kumari, S., & Valakunde, N. D. (2017, December). An IoT based smart solution for leaf disease detection. In *2017 International Conference on Big Data, IoT and Data Science (BID)* (pp. 193-198). IEEE.

Torabi, S., Bou-Harb, E., Assi, C., Karbab, E. B., Boukhtouta, A., & Debbabi, M. (2020). Inferring and investigating IoT-generated scanning campaigns targeting a large network telescope. *IEEE Transactions on Dependable and Secure Computing*.

Trilles, S., González-Pérez, A., & Huerta, J. (2018). A comprehensive IoT node proposal using open hardware. A smart farming use case to monitor vineyards. *Electronics*, *7*(12), 419.

Uddin, M. A., Mansour, A., Jeune, D. L., Ayaz, M., & Aggoune, E. H. M. (2018). UAV-assisted dynamic clustering of wireless sensor networks for crop health monitoring. *Sensors*, *18*(2), 555.

United Nations Economic Commision for Africa (UNECA). (2023). Africa Digital Identity Landscape 2022. Retrieved from https://www.uneca.org/sites/default/files/DITE-AFRICA/Africa%20Digtial%20ID%20Landscape%20Report%20%282023%29.pdf

Vadlamudi, S. (2020). Internet of Things (IoT) in Agriculture: The Idea of Making the Fields Talk. *Engineering International*, *8*(2), 87-100.

Verdouw, C., Sundmaeker, H., Tekinerdogan, B., Conzon, D., & Montanaro, T. (2019). Architecture framework of IoT-based food and farm systems: A multiple case study. *Computers and Electronics in Agriculture, 165*, 104939.

Vodacom, (2023, December 17). Vodacom's innovative solutions improve the delivery of healthcare to those who need it most. Retrieved from https://now.vodacom.co.za/article/future-healthcare-now

World Economic Forum. (2020, January 21). Shaping the Future of the New Economy and Society: A Framework for Action. Retrieved from https://www3.weforum.org/docs/WEF_Annual_Report_2020_21.pdf

Xu, B., Jia, S., Lin, J., Zheng, F., Ma, Y., Liu, L…. & Song, L. (2023, September). JWTKey: Automatic Cryptographic Vulnerability Detection in JWT Applications. In *European Symposium on Research in Computer Security* (pp. 263-282). Cham: Springer Nature Switzerland.

APPENDICES

Appendix I: Conference Presentation Certificate

CERTIFICATE

This is to certify that the following paper was presented online during the
16th International Conference on Information Assurance and Security (IAS 2020)
held on the world wide web during December 15 - 18 , 2020.

Paper ID: 30

Paper Title: MAAMSIC: Multimodal Authentication and Authorization Model For Security of IoT Communication via GSM Messaging in Sub-Saharan Africa

List of Authors: Joan Nabusoba, Calvins Otieno and Wilson Cheruiyot

Springer

Prof. Dr. Ajith Abraham
IAS 2020 - General Chair

Issued On: December 17, 2020

Appendix II: Source Code

```
const express = require('express')



const client = express()



const FinancialTransactions = require('./model/financialModel')

const User = require('./model/userModel')

const IoTDevice = require('./model/IoTDeviceModel')



const mongoose = require('mongoose');



app.use(express.json()) // for parsing application/json

app.use(express.urlencoded({ extended: true })) // for parsing application/x-www-form-urlencoded



//Connect to online db

mongoose.connect('mongodb+srv://cheza-muziki:shikwekwe@cheza-muziki-
y7y2z.mongodb.net/test?retryWrites=true&w=majority', {useNewUrlParser: true});

//initialize db

const db = mongoose.connection;


//redirection endpoint

client.get('/oauthExample/callback', (req, res) => {

  var data = require('querystring').stringify({

    client_id: "clientidclientid",

    client_secret: "clientsecretclientsecret",

    grant_type: "authorization_code",
```

```
    redirect_uri: "http://localhost:8080/",
    code: req.query.code
  })
  var options = {
    uri: "http://localhost:8081/oauth/token",
    body: data,
    method: 'POST',
    headers: {
      'content-type': 'application/x-www-form-urlencoded',
    }
  }
  var request = require('request')
  request(options, (err, response) => {
    if(err) console.log(err);
    else res.send(response.body)
  })
})


app.get('/', (req, res) => {
  let IoTDevice = await IoTDevice.find();


  if (IoTDevice){
      res.status(200).json({ error:false,msg: IoTDevice })
  }else{
      res.status(200).json({ error:true,msg: 'An Error Occured '})
  }
```

```
  res.status(200).json({ error:true,msg:'Data requested cannot be found' })
})

app.get('iot/:id/edit-profile', function (req, res, next) {

  let user = await User.findOne({_id: req.params.id});

  if (user){

     res.status(200).json({ error:false,msg: user })
  }else{
     res.status(200).json({ error:true,msg: 'An Error Occured '})
  }

  res.status(200).json({ error:true,msg:'Data requested cannot be found' })
})

app.get('iot/:id/transactions', function (req, res, next) {
  // req.file is the `photo` file
  // req.body will hold the text fields, if there were any
  let transactions = await FinancialTransactions.find({userId: req.params.id});
  if (transactions){
     res.status(200).json({ error:false,msg: transactions })
  }else{
```

```
    res.status(200).json({ error:true,msg: 'An Error Occured '})
  }
})


app.post('iot/:id/withdraw/:tx_id', function (req, res, next) {
  // req.file is the `photo` file
  // req.body will hold the text fields, if there were any

  let tx = await FinancialTransactions.findOne({_id: req.params.tx_id});

  if (tx){
    const transaction = new FinancialTransactions({
      amount:  tx.amount,
      txDate: tx.txDate,
      status: req.body.status,
      userId: tx.userId,
    });

    await FinancialTransactions.findByIdAndUpdate(req.params.tx_id, transaction);

    res.status(200).json({ error:false,msg: tx })
  }else{
    res.status(200).json({ error:true,msg: 'An Error Occured '})
  }
})
```

```
app.get('iot/:id/IoTDevice', function (req, res, next) {
  // req.file is the `photo` file
  // req.body will hold the text fields, if there were any
  let IoTDevice = await IoTDevice.find({iotId: req.params.id});
  if (IoTDevice){
     res.status(200).json({ error:false,msg: IoTDevice })
  }else{
     res.status(200).json({ error:true,msg: 'An Error Occured '})
  }
})

module.exports = client
```

```
const express = require('express')

const client = express()

const FinancialTransactions = require('./model/financialModel')
const User = require('./model/userModel')
const IoTDevice = require('./model/IoTDeviceModel')

const mongoose = require('mongoose');
```

```
app.use(express.json()) // for parsing application/json

app.use(express.urlencoded({ extended: true })) // for parsing application/x-www-form-urlencoded



//Connect to online db

mongoose.connect('mongodb+srv://cheza-muziki:shikwekwe@cheza-muziki-
y7y2z.mongodb.net/test?retryWrites=true&w=majority', {useNewUrlParser: true});

//initialize db

const db = mongoose.connection;



// const IoTDevice = mongoose.model('IoTDevice', { name: String , releaseDate: String, albumName: Stri
ng, iotId: String, photo: String});

// const FinancialTransactions = mongoose.model('FinancialTransactions', { amount: Number , txDate: Da
te, status: String,userId: String});




//redirection endpoint

client.get('/oauthExample/callback', (req, res) => {

  var data = require('querystring').stringify({

    client_id: "clientidclientid",

    client_secret: "clientsecretclientsecret",

    grant_type: "authorization_code",

    redirect_uri: "http://localhost:8080/",

    code: req.query.code

  })

  var options = {

    uri: "http://localhost:8081/oauth/token",

    body: data,
```

```
    method: 'POST',
    headers: {
      'content-type': 'application/x-www-form-urlencoded',
    }
  }
  var request = require('request')
  request(options, (err, response) => {
    if(err) console.log(err);
    else res.send(response.body)
  })
})


app.get('/', (req, res) => {
  let IoTDevice = await IoTDevice.find();


  if (IoTDevice){
      res.status(200).json({ error:false,msg: IoTDevice })
  }else{
      res.status(200).json({ error:true,msg: 'An Error Occured '})
  }


  res.status(200).json({ error:true,msg:'Data requested cannot be found' })
})
```

```
app.get('iot/:id/edit-profile', function (req, res, next) {


  let user = await User.findOne({_id: req.params.id});


  if (user){




     res.status(200).json({ error:false,msg: user })

  }else{

     res.status(200).json({ error:true,msg: 'An Error Occured '})

  }


  res.status(200).json({ error:true,msg:'Data requested cannot be found' })

})


app.get('iot/:id/transactions', function (req, res, next) {

  // req.file is the `photo` file

  // req.body will hold the text fields, if there were any

  let transactions = await FinancialTransactions.find({userId: req.params.id});

  if (transactions){

     res.status(200).json({ error:false,msg: transactions })

  }else{

     res.status(200).json({ error:true,msg: 'An Error Occured '})

  }
```

```
})



app.post('iot/:id/withdraw/:tx_id', function (req, res, next) {
  // req.file is the `photo` file
  // req.body will hold the text fields, if there were any

  let tx = await FinancialTransactions.findOne({_id: req.params.tx_id});

  if (tx){
     const transaction = new FinancialTransactions({
        amount:  tx.amount,
        txDate: tx.txDate,
        status: req.body.status,
        userId: tx.userId,
     });

     await FinancialTransactions.findByIdAndUpdate(req.params.tx_id, transaction);

     res.status(200).json({ error:false,msg: tx })
  }else{
     res.status(200).json({ error:true,msg: 'An Error Occured '})
  }
})
```

```
app.get('iot/:id/IoTDevice', function (req, res, next) {

  // req.file is the `photo` file

  // req.body will hold the text fields, if there were any

  let IoTDevice = await IoTDevice.find({iotId: req.params.id});

  if (IoTDevice){

      res.status(200).json({ error:false,msg: IoTDevice })

  }else{

      res.status(200).json({ error:true,msg: 'An Error Occured '})

  }

})



module.exports = client
```

Flutter App

```
import 'package:flutter/material.dart';

import 'package:maamsic/screens/intro.dart';

import 'package:maamsic/screens/messages.dart';

import 'package:maamsic/screens/sign_in.dart';

import 'package:maamsic/screens/device_list.dart';

import 'package:maamsic/uidata.dart';


void main() {

 runApp(MyApp());

}


class MyApp extends StatelessWidget {

  // This widget is the root of your application.

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      debugShowCheckedModeBanner: false,

      title: 'MAAMSIC',

      theme: ThemeData(

        primarySwatch: Colors.blue,

      ),

      home: Intro(),
```

```
      routes: {
        UIData.introRoute: (BuildContext context) => Intro(),
        UIData.signInRoute: (BuildContext context) => SignIn(),
        UIData.deviceListRoute: (BuildContext context) => DeviceList(),
        UIData.messagesRoute: (BuildContext context) => Messages(),
      },
    );
  }
}
```

```
import 'package:flutter/material.dart';
import 'package:maamsic/uidata.dart';


class Intro extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        backgroundColor: UIData.mainColor,
        body: Stack(
          children: <Widget>[
```

```
            NestedScrollView(
                headerSliverBuilder:
                    (BuildContext context, bool innerBoxIsScrolled) {
                  return <Widget>[
                    SliverAppBar(
                      backgroundColor: UIData.mainColor,
                      expandedHeight: 200.0,
                      floating: false,
                      pinned: true,
                      flexibleSpace: FlexibleSpaceBar(
                          centerTitle: false,
                          titlePadding: EdgeInsets.zero,
                          title: Container(
                            padding: EdgeInsets.only(left: 16, bottom: 16)
,
                            child: Text("MAAMSIC",
                                style: TextStyle(
                                  color: Colors.white,
                                  fontSize: 16.0,
                                )),
                          ),
                          background: ShaderMask(
                            shaderCallback: (rect) {
```

```
                            return LinearGradient(
                              begin: Alignment.topCenter,
                              end: Alignment.bottomCenter,
                              colors: [Colors.black, Colors.transparent]
,
                            ).createShader(
                                Rect.fromLTRB(0, 0, rect.width, rect.hei
ght));
                          },
                          blendMode: BlendMode.dstIn,
                          child: Image.asset(
                            "assets/img/banner.jpg",
                            fit: BoxFit.cover,
                          ),
                        )),
                    ),
                  ];
                },
                body: SingleChildScrollView(
                  child: Container(
                    padding: EdgeInsets.only(top: 20, left: 20, right: 20)
,
                    child: Column(
```

```
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: <Widget>[
                      Text(
                        "Synopsis",
                        style: UIData.heading,
                      ),
                      SizedBox(height: 16),
                      Text(UIData.introDesc, style: UIData.normal),
                    ]),
                ),
            )),
        Positioned(
            bottom: 0,
            left: 0,
            right: 0,
            child: Container(
                color: UIData.mainLightColor,
                padding: EdgeInsets.symmetric(vertical: 10, horizontal
: 20),
                height: 65,
                child: Row(
                    mainAxisAlignment: MainAxisAlignment.spaceBetween,
                    children: <Widget>[
```

```
                        Expanded(
                          child: FlatButton(
                              shape: RoundedRectangleBorder(
                                borderRadius: BorderRadius.circular(10.0
),
                              ),
                              color: UIData.orangeColor,
                              onPressed: () {
                                Navigator.pushNamed(
                                    context, UIData.signInRoute);
                              },
                              child: Text("MAAMSIC Login",
                                  style: UIData.buttonText)),
                        ),
                        SizedBox(width: 10),
                        Expanded(
                          child: FlatButton(
                              shape: RoundedRectangleBorder(
                                  borderRadius: BorderRadius.circular(10
.0),
                                  side:
                                      BorderSide(color: UIData.orangeCol
or)),
```

```
                                onPressed: () {
                                  Navigator.pushNamed(
                                      context, UIData.signInRoute);
                                },
                                child: Text("Malicious Login",
                                    style: TextStyle(
                                        color: Colors.white60,
                                        fontWeight: FontWeight.bold,
                                        fontSize: 15))),
                            )
                        ]))
                ],
              ),
            ),
        );
    }
}
```