Performance Analysis of Pharmaceutical Cold Chain Monitoring on ESP32: Encrypting Thermal Data Using the NIST Lightweight Cryptography Standard (Ascon)

By

Sikder Mohammad Mahdi Swad
22101126
Maliha Hossain Munia
22101115
Raiyan Ibn Taher
21241026

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
BRAC University
January 2026

Declaration

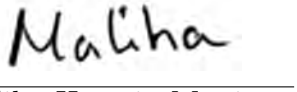
It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at BRAC University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Sikder Mohammad Mahdi Swad
22101126

Student's Full Name & Signature:



Maliha Hossain Munia
22101115

Raiyan Ibn Taher
21241026

Approval

The thesis/project titled "Performance Analysis of Pharmaceutical Cold Chain Monitoring on ESP32: Encrypting Thermal Data Using the NIST Lightweight Cryptography Standard (Ascon)" submitted by

1. Sikder Mohammad Mahdi Swad (22101126)
2. Maliha Hossain Munia (22101115)
3. Raiyan Ibn Taher (21241026)

Of Fall, 2026 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on January 31, 2026.

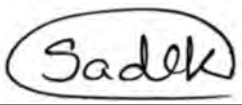
Examining Committee:

Supervisor:
(Member)

Partha Bhoumik

Lecturer
Department of Computer Science and Engineering
BRAC University

Co-Supervisor:
(Member)



Dr. Md Sadek ferdous

Professor
Department of Computer Science and Engineering
BRAC University

Thesis Coordinator:
(Member)

Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Chairperson
Department of Computer Science and Engineering
BRAC University

Abstract

In modern medical logistics and pharmaceutical systems, temperature monitoring of medicines is a crucial factor that ensures drug quality and safety. Often, this sensitive data is vulnerable to cyberattacks and unauthorized access when transmitted or stored digitally. The aim is to develop a secure temperature encryption and decryption system for medicine datasets using an ESP32 microcontroller. The primary objective is to compare the encryption time and performance between ASCON [1], AES [2], ChaCha-20 Poly1305 [3], TinyJAMBU [4], and GIFT-COFB [3] using ESP32. The methodology includes collecting a dataset containing medicine-related attributes, extracting the temperature column, applying the chosen encryption—NIST Lightweight Cryptography Standard (ASCON) [1], AES-GCM (Advanced Encryption Standard in Galois/Counter Mode) [2], ChaCha20-Poly1305 [3], TinyJAMBU [4], and GIFT-COFB [3]—via ESP32, analyzing performance metrics including encryption time and computational efficiency. After encryption, we design a prototype with a DHT22 sensor [5] that will detect the temperature data and work with ESP32 to encrypt the data, sending the data to a verified person. For this, we design a protocol that provides replay protection and device authentication [6] so that medicines are safer from targeted attackers. The contribution is a measurement of encryption latency, code size, and power consumption, showing that ASCON achieves significantly higher performance and energy efficiency while maintaining adequate security, making it more suitable for constrained IoT devices [1]. Furthermore, we remove the printstate function to run in lesser time while also removing the debugging setup in anyday use.

Keywords: IoT Security; ESP32; ASCON; Data Encryption; Medicine Dataset; Performance Analysis; Temperature Data; Cryptography.

Acknowledgement

Firstly, we want to thank ALLAH for allowing us to work on this which can mitigate security problem in iot sector of coldchain storage and relative works .
Secondly, our supervisor Partha Bhoumik sir and our co-supervisor Dr. Md Sadek Ferdous sir for their support and advice throughout the whole thesis phase.
Thirdly, in the poster presentation, where we have been selected, our BRAC University faculties give us necessary advices to continue our work.
Infine, to our parents who keep supporting us and pray for us.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Overview

Cryptography is essential to the security and privacy of medical data in the digital era. Cryptographic techniques offer the crucial defense mechanisms that ensure authenticity, secrecy, and integrity in all digital communications and data storage systems through the encryption and decryption processes [1]. The Internet of Things (IoT) ecosystem's rapid expansion poses a unique challenge to this reliance on secure communication in the pharmaceutical supply chain [7]. Microcontrollers and sensors are among the many devices used in Internet of Things deployments that have severe resource limits, such as low computing power, small random-access memory (RAM), and a restricted energy supply. The performance and operational lifespan of these resource-constrained devices may be jeopardized by the substantial computational and temporal overheads that standard cryptographic methods frequently bring, notwithstanding their high level of security. This performance-security trade-off needs to be addressed for the IoT to be used safely and widely [8]. To mitigate these challenges, lightweight cryptographic algorithms designed specifically for restricted situations have become increasingly important [4]. In this case, authenticated encryption systems such as ASCON provide an efficient solution by simultaneously ensuring confidentiality, integrity, and authenticity with the least amount of computational cost [1]. In order to prevent replay attacks and device identity as attached data, the proposed protocol uses ASCON based Authenticated Encryption with Attached Data (AEAD), which includes a counter-based nonce technique. By creating per message session keys and lowering memory and processing requirements, the protocol enables secure sensor-to-server connection while being practical for low power IoT devices used in the pharmaceutical supply chain.

Study Purpose

The purpose of this study is to make a prototype with a protocol which is lightweight and secure [7]. This study will specifically analyze important performance indicators during NIST Lightweight Cryptography Standard (ASCON) in encryption and decryption, including execution time, RAM use, and power efficiency [1]. An ESP32 microcontroller loaded with lightweight implementations will be used to compare a resource-constrained environment with a traditional computing environment which is a regular Windows operating system. An innovative aspect of this investigation is the assessment of the ESP32's performance in five encryption models [1], [4], [8], [2], [3]. Furthermore, its performance with DHT22 sensor to encrypt and decrypt with defined protocol will create a new vision in the world [5]. The analysis's conclusions will offer insightful data that will help choose the best cryptographic techniques for systems with limited resources in the future.

1.1 Background

The way of monitoring and storing temperature sensitive goods over the cold chain has evolved due to the growing reliance on the Internet of Things (IoT) by the pharmaceutical industry [7]. Temperature monitoring and protection of data throughout the manufacturing process to delivery is of critical importance in ensuring that the medications are not rendered ineffective [7]. The ESP32 microcontroller, chosen in this research, because of its low power consumption, built-in wifi and Bluetooth, and effective computing power, will be assembled into a prototype Internet of Things system [2]. The technology tries to provide a secure real time monitoring application that can help protect pharmaceutical temperature information against unauthorized access and also maintain system performance by combining temperature detection with a lightweight encryption algorithm [1]. The ESP32 will be coded using the Arduino IDE to collect the temperature readings of the sensors then encrypt and broadcast it via wifi to a cloud platform where it will be stored and immediately analyzed using cloud services [3]. We will also work on an interface that would be utilized to show a temperature reading and allow users to remotely monitor the temperature [7]. The accuracy, security, and general performance of the sensor will be the primary emphasis of the initial testing [5].

1.2 Research Motivation

Addressing the Performance–Security Trade-off

A fundamental issue has been brought about by the quick adoption of the Internet of Things (IoT) in pharmaceutical logistics, such as the use of microcontrollers like the

ESP32 to monitor the vital cold chain [7]. Even though they provide the highest level of security, traditional cryptographic algorithms need a huge amount of resources. Significant concessions are made when they are used on small, battery-powered devices, mainly because they introduce high latency and deplete limited energy reserves, shortening operating lifetime [1]. Practitioners must decide whether to sacrifice device performance for security or security for performance as a result of this trade-off. By explicitly comparing the performance of the ASCON encryption process to that of the other encryption processes AES-GCM (Advanced Encryption Standard in Galois/Counter Mode) [2], ChaCha20-Poly1305 [3], TinyJAMBU [4], and GIFT-COFB [3] encryption processes in ESP32, our work aims to quantify this specific overhead. By monitoring the variations in time, RAM, and power efficiency under different temperature circumstances, this work will develop data-driven benchmarks to guide the selection and development of lightweight cryptographic algorithms [8]. This will allow for strong security without sacrificing the effectiveness of life-saving medication monitoring devices [1].

Practical Validation of Lightweight Cryptography

The theoretical benefits of lightweight cryptographic primitives have been shown in academic literature, but their practical performance under environmental stress is unknown [1]. This work will be driven by the necessity to verify performance measures of embedded systems, and it will offer a new analysis through the description of temperature sensitivity of ASCON encryption and decryption on the ESP32. The transportation of pharmaceuticals is often subject to changes in temperature, and this may affect the security in an unpredictable way because they affect the power consumption and processing performance of microcontrollers [7].This work seeks to fill in the gap between theoretical efficiency of cryptography and practical system engineering by demonstrating the impact of heat conditions on the cost of security [3]. This will provide meaningful information in the future development of cryptographic firmware and hardware in critical Internet of Things applications [1].

1.3 Problem Statement

Throughout the transportation process of vaccines and medications, the pharmaceutical cold-chain is important to maintain the integrity of temperature-sensitive products [7]. Ensuring the reliability and security of data is one of the challenges as any unauthorized access or tampering can significantly compromise the integrity of pharmaceutical products. Traditional temperature monitoring systems lack strong security measures, making them very vulnerable to cyberattacks [1]. In this research, we will be developing an IoT system using ESP32, which encrypts the temperature, ensuring temperature information

is secure and protected from potential attackers who might want to jeopardize the quality of medicine for certain reasons [1]. Furthermore, this research also aims to develop a protocol to intentionally increase the time for the decryption process and hence enhance the security of this data. Due to the implementation of this protocol, not only will this be time-consuming, it will also require strong computational power to decrypt, thus maintaining the integrity of the pharmaceutical products [8].

1.4 Objectives (Research Goals)

This study's primary goal is to create a secure ESP32-based Internet of Things (IoT) device for real-time temperature monitoring with safe data transfer. To increase resistance against unauthorized access, a session-based protocol is added, and strong encryption mechanisms are used to protect the collected data.

1. **RO1: Analyzing Current Temperature Data Encryption Mechanisms** — We initially look into and explore existing monitoring, authentication, and encryption mechanisms which might be used in similar IoT solutions [1]. This will involve reviewing their security weaknesses, performance, and any form of restriction, and with this analysis, we will suggest more efficient ways of encrypting data with the help of the ESP32-based system to ensure that their data is not prone to any unauthorized access [2].

2. **RO2: Developing a Secure ESP32-Based System** — We design and develop an ESP32-based IoT device that shall sense the temperature data at the cold storage[7]. This system will be inclusive with encryption techniques and its safe transmission. The system will be created in such a way that it will guarantee confidentiality and integrity of the temperature data [1].

3. **RO3: Integration of Temperature Sensors for Data Collection** — We combine a high-precision temperature sensor that will provide us with the real-time temperature reading to monitor the temperature [5]. The sensor will allow a stable and quality provision of data that will be encrypted to avoid unauthorized access[1].

4. **RO4: Data Encryption and Decryption for Secured Transmission** — We apply the encryption method to achieve secure transmission of data. Data will be encrypted by use of encryption algorithms like ASCON [1],that will facilitate a smooth transition and will exclude any form of unauthorized access when storing or in transit [1]. Moreover, we will equally be comparing various encryption algorithms including comparing it with AES, in terms of speed and resource consumed [2]. We will also implement a time delay protocol to the decryption process hence making the process more complicated to the hackers attempting to get access to the temperature data [3].

5. **RO5: Testing and Validation** — We put the system to rigorous tests in a controlled setting that confirms the accuracy and robustness of the temperature readings of the encryption techniques in case of cyberattacks [1]. This is aimed at ensuring that this system is effective in a real-life scenario without jeopardizing security.

1.5 Methodology in Brief

The primary objective of the given study is the construction of a prototype of a safe Internet of Things-based system of temperature monitoring. The ESP32 microcontroller which has been selected as a real-time and resource-constrained IoT development has been selected because of its built-in Wi-Fi, bluetooth capabilities, sufficient processing power, and energy-saving features, etc. [2]. An ESP32 is connected to a DHT22 temperature and humidity detector[5] to constantly check ambient variables in the state of nature. The immediate protection of the temperature data collected is performed by the lightweight authenticated encryption mechanism once the data have been structured into a lightweight format of data. Specifically, the system uses the ASCON AEAD (Authenticated Encryption with Associated Data) algorithm, which is a lightweight cryptography approach that is NIST-standard and designed to be utilized by Internet of Things devices [1]. This ensures integrity, confidentiality and validity of data during the process of transmission [1].

The encryption scheme is a nonce-based encryption scheme to prevent replay attacks and unauthorized reuse of data [1]. Each encryption operation is made unique by generating a nonce which is a device specific random number with a monotonically increasing sequence counter. This nonce is permanently stored in non-volatile memory of the ESP32 [1]. Associated data (AD), e.g. device identity and sequence data, are also encrypted so that vital metadata cannot be tampered with without revealing it to encryption[1]. The ESP32 is configured to read sensor data and encrypt the data in real-time to transmit the secure payload to a cloud-based platform using Wi-Fi, by configuring it with Arduino IDE [3]. The encrypted temperature data is stored and processed through a cloud service which can then be analyzed and represented through a user interface to enable remote cold-chain condition monitoring [7].

Additionally, it decrypts the encrypted payload since the decryption protocol is carefully designed to add computational delay [1]. This architecture option improves security by discouraging brute-force and unlawful decryption efforts, especially when encrypted data might be captured along the way. It can only be decrypted using the appropriate cryptographic key, nonce and associated data due to the strict control over access [1]. The main areas of interest of system evaluation are sensor precision, cryptographic functions, and the overall system integrity, such as replaying and tampering vulnerability and encryption/decryption delay. This approach also shows that real-time IoT sensing and

lightweight encryption could be used to ensure robust and safe temperature monitoring of cold-chain systems [1].

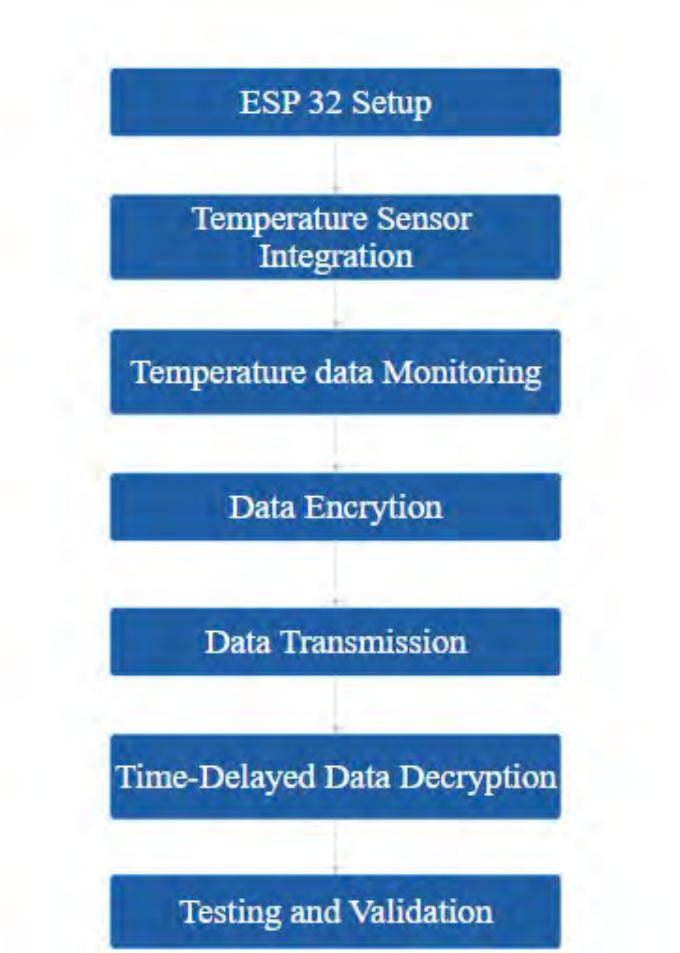


Figure 1.1: Flowchart of the Proposed Work

1.6 Scopes and Challenges

Scope: This study narrows down to consider the encryption of temperature data. The key goal is to encrypt the temperature data to avoid unauthorized access to prevent pharmaceutical products integrity [7]. In this paper, we will take into account five encryption protocols, which are ASCON[1], AES-GCM (Advanced Encryption Standard in Galois/Counter Mode) [2], ChaCha20-Poly1305 [3], TinyJAMBU [4], and GIFT-COFB [3] in ESP32 microcontroller. This research will focus on speed and efficiency of encryption. Also, the design and testing of time-delay decryption protocol will be pursued[1].The paper will compare the algorithm and its speed of processing transactions with both encryption protocols on the parameters of real-time process rate and efficiency and the way the decryption protocol provides an extra level of security [2].

Challenges: In the process of this research, there was a fair likelihood of meeting some challenges. To begin with, the ESP32 could be limited to run encryptions efficiently due to its low computational resources [2]. Furthermore, there might be difficulties in comparing the performance of different encryption algorithms due to their differing computational requirements. Additionally, the incorporation of libraries into embedded.systems might be in need of optimization to achieve efficiency in the use of resources, and it will also be necessary to make sure that encryption strength is not compromised [3]. We must see whether our time-delay decryption scheme would impose any substantially noticeable burden on the already limited computing capabilities of the ESP32, or otherwise burden our authorized users in any respect. Finally, it is significant that both platforms should have the same conditions and environments when conducting tests in order to get reliable and comparable results [7].

1.7 Report Structure

The format of this report is separated into several chapters, which represent the significant aspects of the process of research and which present a detailed picture of the methodology, findings and conclusions. The chapters are as follows:

1.7.1 Chapter 1: Introduction

In this chapter, the overall background and significance of the study are presented. History of temperature-sensitive drugs and the challenges involved in maintaining their integrity during storing and transportation are discussed [7]. The need to use safe and effective data encryption when it comes to the cold-chain monitoring systems in the Internet of Things is justified by the research motivation [1]. This chapter provides the description of the issue, specific objectives of the study, and methods applied in it. It closes by giving details on the scope of the research and the challenges that are likely to be experienced during the project in order to prepare the groundwork on the intensive discussions in the subsequent chapters [1].

1.7.2 Chapter 2: Literature Review

This chapter provides a detailed literature review of the field of IoT security, temperature monitoring and cryptography, with the focus on resource-constrained solutions due to the need to create lightweight cryptography that suits the internet of things devices [1]. The chapter reviews the previous studies on the cryptography of temperature data, modern cryptographic protocols, and the specific techniques used in the application. It also considers the NIST Lightweight Cryptography (LWC) standard and evaluates whether

various encryption techniques are suitable in embedded systems [1]. The review provides the background to this research study by identifying voids in the literature especially in terms of performance analysis and implementation of lightweight cryptography to the Internet of Things systems [8].

1.7.3 Chapter 3: Requirements, Impacts, and Constraints

This chapter outlines the specific requirements of the research venture such as the hardware and software requirements. The necessary hardware components are the ESP32 microcontroller and sensor, namely, DHT22 which is used in the monitoring of real-time temperature [5]. One of the software concerns is to implement lightweight cryptographic techniques to ensure safety of data without compromising system performance [1]. This chapter also discusses the sociological, environmental and ethical implications of the research and pays special attention to the importance of preventing temperature data leakage to the population, reducing the pharmaceutical waste and ensuring compliance with the security laws [7]. The last part of the chapter is the discussion of risk management and economic analysis, which will evaluate the scalability and cost-effectiveness of the proposed system [1]. The constraint of interacting with resource limited devices is discussed [2].

1.7.4 Chapter 4: Methodology

Chapter 4 provides a detailed description of the processes involved in building, constructing, and evaluating the secure temperature monitoring system and it gives a detailed description of the research methodology [1]. The three design processes are data preparation, system implementation, and performance evaluation. The methods used to collect and preprocess data to simulate real temperature data and prepare it to be encrypted are discussed in this chapter. It also explains implementation of the system such as implementation of various lightweight encryption method, implementation of ESP32 and implementation of the DHT22 sensor [1]. This chapter also discusses the security measure employed to ensure integrity of data and prevent unauthorized access. Also, there is the provision of a paradigm upon which performance is evaluated and this paradigm compares the encryption and decryption performance of the various algorithms under the same testing conditions [1].

1.7.5 Chapter 5: Result Analysis

This chapter presents the experiments that were conducted to determine the effectiveness of the cryptographic algorithms applied in the Internet of Things system. Results are analysed using key performance parameters which include encryption delay, throughput,

power consumption, RAM usage and energy efficiency. The five AEAD (Authenticated Encryption with Associated Data) ciphers, which include ASCON-128 [1], AES-128 GCM [2], ChaCha20-Poly1305 [3], and TinyJAMBU [4], are contrasted in the chapter, along with their strengths and weaknesses in terms of their applicability to IoT devices with limited resources [1]. The validity of the findings is provided with the discussion of statistical methods that were employed to be sure of the reliability of the outcomes. This chapter discusses the best algorithms to use in the safe and effective temperature monitoring in cold-chain applications [1].

1.7.6 Chapter 6: Conclusion

The key conclusions of the research are presented in the final chapter, which also derives conclusions of the analysis of the performance and comparison of encryption methods [1]. It discusses the contributions of the study to the security of the IoT, particularly in application to cold-chain monitoring of pharmaceuticals [7]. The optimization of the performance, the practice of applying them to an industrial environment in an internet of things, and the use of machine learning models with encrypted data are only a few future research opportunities that can be determined in this chapter [1]. The significance of lightweight cryptography in securing IoT systems is listed in the conclusion that also outlines how the study has influenced the design of implementable, scalable IoT security systems [1].

Chapter 2

Literature Review

2.1 Preliminaries

- **Encryption/Cryptography:** Cryptography is the science of ensuring confidentiality, integrity and authenticity of data by using mathematical algorithms. It involves data transformation into a secure form that could be read or seen only by the people having the right kind of decryption key or credentials. Encryption is the process by which an encryption algorithm can convert readable data (plaintext) into unreadable data (ciphertext) and is a key element of cryptography. The main aim of encryption is to secure sensitive data during the process of transmission and storage.

  Due to the relatively poor security facilities of resource limited devices such as microcontrollers, encryption is necessary in the Internet of Things (IoT) environment, including the pharmaceutical cold chain monitoring system, which is used. These gadgets are often employed in environments with limited memory, processing capabilities and battery life and it is very hard to effectively apply traditional encryption algorithms. Due to this, lightweight cryptographic algorithms, which are designed to provide a high level of security without causing serious processing capabilities, have become more important in securing the Internet of Things devices.

  To avoid unauthorized access, manipulation, and cyberattacks, lightweight encryption solutions can be used to safeguard sensitive information including temperature data of pharmaceutical products. Indicatively, special purpose algorithms such as ASCON have been developed specifically to be used in an IoT setting to ensure that information can be encrypted securely and efficiency maintained in systems that have minimal resources. ASCON, as a part of the NIST Lightweight Cryptography standard, offers a lightweight authentication encryption with associated data (AEAD) with a guarantee of integrity and confidentiality of data and minimal processing power [1].

  The biggest challenge in applying cryptographic algorithms in the IoT system particularly in the real-time temperature supervision in the pharmaceutical system is bal-

ancing system efficiency and security. Although they are extremely safe, conventional encryption protocols such as AES (Advanced Encryption Standard) require a significant amount of processing power and memory, and low-power microcontrollers such as the ESP32 do not necessarily have this. As an example, among the most popular encryption algorithms used in secure communications is AES-GCM (AES in Galois/Counter Mode) but due to its high processing power, it might be inefficient in an embedded system [2].

On the other hand, AEAD algorithm ChaCha20-Poly1305 is another lightweight encryption algorithm that has been proven to be secure and effective even in situations where there are limited resources [3]. ChaCha20 is a stream cypher that provides excellent encryption and integrity as well as validity of the encrypted data when used with Poly1305 message authentication code. It is due to this reason that ChaCha20-Poly1305 is an ideal choice when it comes to securing the Internet of Things systems where bandwidth, power and computing capacity are major concerns.

Similarly, other algorithms such as TinyJAMBU and GIFT-COFB can be used in place of lightweight cryptography in Internet of Things devices. TinyJAMBU is a high throughput and low-latency cryptography library, as part of the NIST Lightweight Cryptography project, so that it can be used in time-sensitive applications, including monitoring temperature-sensitive pharmaceuticals [4]. A third cryptographic algorithm, GIFT-COFB, is another lightweight algorithm that is applicable to the pharmaceutical cold chain IoT systems [3].

In addition to encryption techniques, there are such techniques as nonce-based encryption schemes applied to prevent replay attacks and ensure that previously transmitted data cannot be reused or tampered with in order to make the encryption process efficient and secure in IoT systems. A nonce is some random number which is used once only in a cryptographic connection to prevent the reuse by a hacker of intercepted messages. The technique can be used with minimal power usage and processor overhead [1] when implemented on microcontrollers like the ESP32 and enhances the security of the data. Authenticated Encryption with Associated Data (AEAD) is an essential element of encryption in Internet of Things systems ensuring the secrecy and authenticity of data, as well as the integrity. AEAD schemes such as ASCON are another security layer of an IoT system that encrypts the data as well as ensuring that it has not been altered in transit. This is particularly important in sensitive environments such as pharmaceuticals where the consequences of manipulating data can prove to be very detrimental [1].

Also, in IoT based systems, encrypted data collection in real time alongside data collection is critical. Indicatively, to avoid unwanted access, it is important that temperature data received by sensors should be encrypted immediately. DHT22 temperature sensor

real-time temperature measurements require that the information should be encrypted immediately and then sent to a cloud system. This ensures that the data is secure in all the cold-chain processes, such as data collection up to cloud storage and analysis [5].

Finally, energy efficiency is also an important consideration in the implementation of cryptography of the IoT system. IoT devices also have a small amount of power, so one should not choose encryption methods that drain the battery badly. Lightweight encryption algorithms like ASCON, ChaCha20-Poly1305, and TinyJAMBU are energy-efficient and provide strong encryption without consuming a lot of power, which is essential in the long-term implementation in the IoT systems, especially in such environments as pharmaceutical cold chain monitoring where the devices have to run continuously without a change in batteries [7].

To sum up, the security of Internet of Things systems cannot exist without cryptography particularly in areas like cold chain monitoring of pharmaceuticals. IOT devices may provide effective data security and maintain the efficiency in resource-limiting scenarios with the help of lightweight cryptographic algorithms ASCON, AES-GCM, ChaCha20-Poly1305, TinyJAMBU, and GIFT-COFB. These are lightweight encryption techniques that provide the IoT devices with the capability to operate safely and efficiently in the real world by balancing the challenges of creating a balance among security, performance, battery use, and real-time processing requirements.

- **ESP32: Microcontroller and its Hardware Features**

  The ESP32 is a low-power, popular, and inexpensive microcontroller that offers significant performance advantages on Internet of Things (IoT) applications. One of the key aspects of the ESP32 that render it ideal in real-time data transmission within the networked system such as the pharmaceutical cold chain monitoring system addressed in this paper citeBuchanan2017 is its built-in Wi-Fi and Bluetooth functions. The ESP32 is well adapted to the Internet of Things environment where resource limitations, i.e., power, memory, and processing rate are typical due to the integration of wireless communication technologies that allow the ESP32 to implement data collecting, processing, and safe transfer in the most efficient way. Performance and Hardware Characteristics The ESP32 has two cores with 32-bit computing capabilities having the capacity to run other processes and handle complex functions in real time at a clock speed of up to 240 MHz [7]. The use of lightweight cryptographic algorithms plays a critical role in ensuring the speed of the systems without compromising the security due to the fact that even though it is efficient, this processing power is still far less than the more powerful desktop processors. The ESP32 has deep sleep modes and flexible power management, which significantly increase battery life in Internet of Things applications, such as temperature monitoring systems, which require long periods of

operation [2].

To be able to communicate with the numerous sensors and actuators (among other devices), the ESP32 also supports a wide array of peripherals, such as SPI, I2C, UART and PWM. In this study, the ESP32 is interfaced with the temperature sensor, DHT22, that is charged with the responsibility of collecting real-time temperature data in the pharmaceutical cold chain [5]. The sensor data are first encrypted with lightweight cryptographic techniques before being posted to a cloud based server to be stored and analyzed [7]. Bluetooth and Wi-Fi integration
Integration of Bluetooth and Wi-Fi of ESP32 is particularly suitable when it comes to the Internet of Things projects, which require remote monitoring and data transmission due to its Wi-Fi and Bluetooth features. The ESP32 also allows real-time transmission of encrypted temperature data to a cloud-based storage system via the connection to the local and the wide-area network. To prevent any loss in product integrity in the pharmaceutical cold chain monitoring system, the transfer of temperature data through long distances has to be effectively and safely transferred [1]. ESP32 is ideal when dual communication is required in a device since it has the ability to support both Wi-Fi (802.11 b/g/n) and Bluetooth (Classic and BLE) in a single chip that reduces the overall system complexity.

The ESP32 has the capability to safely communicate with remote systems due to the Wi-Fi and Bluetooth modules. The integrated TLS/SSL support ensures that the information which is transmitted over the network is not subject to manipulation and eavesdropping as it is encrypted. Also, physical access to the device might be limited, so the OTA (Over-the-Air) update feature allows remote firmware upgrades, which is particularly useful in cases of long-term deployment[1].

Encryption and Real-Time Data Gathering
The real-time data collection ability of the ESP32 is one of the key components of its role in the present research. The temperature-sensitive products in pharmaceutical logistics must continually be monitored in order to ensure that the products maintain their set limits in temperature. The ESP32 microcontroller can package temperature readings of sensors like DHT22 and immediately encrypt it with the help of basic cryptographic tools. The safety of the encryption does not overload the resources of the system due to the usage of the ASCON and other lightweight AEAD (Authenticated Encryption with Associated Data) algorithms that are specifically designed to operate in limited settings such as ESP32 [1].

In this study, the encryption process is applied to the encryption of real-time temperature data with the help of the ASCON AEAD method that encodes data into an incoherent form (ciphertext). This method makes sure that the temperature data was not tampered with prior to transmission through encrypting and authenticating

it citeSerrano2022. Since ESP32 microcontroller is used in a resource-constrained environment, the choice of lightweight cryptographic protocols, including ASCON, AES-GCM, ChaCha20-Poly1305, and so on, are critical to making sure that the encryption process does not impose any serious delays and power consumption overheads[2], [1], [4]. Power Efficiency and Extended Use

Power Economy and Long Life Time Low power consumption of the ESP32 is among its key features. In the intervals between collecting or sending data, the microcontroller can significantly reduce its power use due to advanced power management features such as deep sleep mode. This is particularly vital in the case of IoT systems such as pharmaceutical cold chain monitoring, where sensors should operate over a long duration of time without any manipulation. ESP32 is also suitable in the long-term applications in resource-constrained environments due to its good power consumption and ability to safely encrypt data on-the-fly [7].

Also, the battery life of ESP32 can also be prolonged by the provision of low-power operation mode of Wi-Fi that does not allow the device to spend too much energy when it is not transmitting data. This is particularly essential with cold-chain applications, where devices are often put in remote or inaccessible locations, and the functionality of the system has a direct influence on the feasibility and affordability of the solution [2]. Authentication and Security

Authentication and Security capability of ESP32 is also a key consideration in designing the system, along with the features of the physical aspect. There are several security protocols that the ESP32 is compatible with, including Secure Boot to ensure that only trusted code is run on the platform as well as the encryption of the SSL/TLS protocols. In the present research, the data-safety capabilities of the ESP32 to encrypt temperature data and send it to remote servers over encrypted data channels are vital in maintaining the integrity and secrecy of the data during the cold chain process [1]. The nonce-based encryption method in the system also enhances the security of the system since this approach prevents replay attacks by ensuring that every encryption activity is unique and cannot be repeated by the attackers[1].

Final Thoughts

Application ESP32 microcontroller forms an ideal base for developing safe, efficient, and real-time Internet of Things systems in resource-constrained environments. The low power consumption and in-built Wi-Fi and Bluetooth features make it a good choice in applications such as pharmaceutical cold chain monitoring where secure transmission, encryption and real-time data collection are very important. By using the lightweight cryptographic protocols such as ASCON, the ESP32 can ensure the security of the critical temperature data without sacrificing the system durability, performance or

power efficiency. This paper proposes an ESP32 as the main component of the Internet of Things (IoT)-based temperature monitoring system as it has the computational ability, communication, and security features [7], [1], [2] due to its combination of computational power, communication capabilities, and security features.

- **ASCON-128 and ASCON-128a:** ASCON-128 and ASCON-128a were coded in the `ascon_aead128_encrypt`, `ascon_aead128a_encrypt`, `ascon_aead128_decrypt`, and `ascon_aead128a_decrypt` functions according to the NIST Lightweight Cryptography (LWC) standard. These are designed with low memory footprint and optimized performance on resource-constrained hardware while providing secure sensor data encryption and authentication using a single AEAD interface.

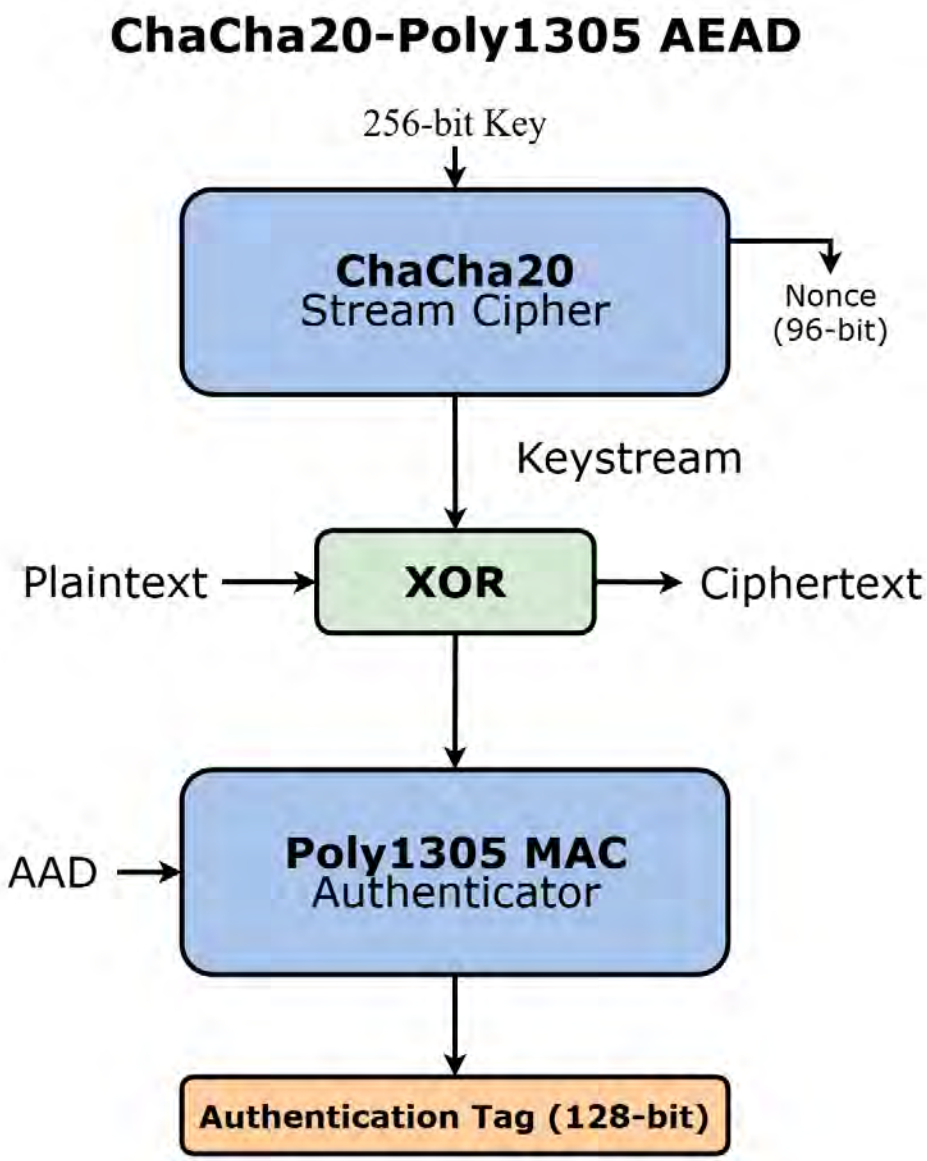


Figure 2.1: Architectural overview of ChaCha10-Poly1305 AEAD encryption workflows

- **AES-128 GCM:** AES-128GCMwas implemented on the mbedTLS library using the in-built ESP32 hard ware AES accelerator. The encryption and the authentication functions were performed with the mbedtls-gcm-encrypt-and-tag and mbedtls-gcm-auth decrypt functions. This model provides high throughput and uses the same nonce

generation and related data (AAD) authentication logic as the lightweight ciphers as the performance benchmark baseline.

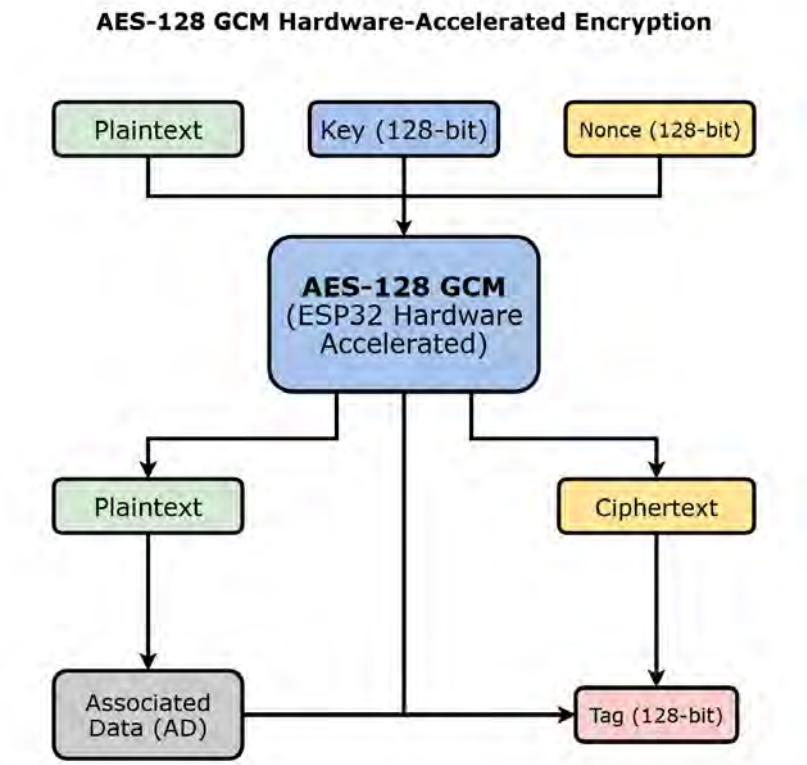


Figure 2.2: Architectural overview of AES-128 GCM hardware-accelerated encryption workflows

- **ChaCha20-Poly1305:** ChaCha20-Poly1305 AEAD cipher was coded using the Arduino Crypto library via the ChaChaPoly interface. Every message was sent with a 256-bit secret key and 96-bit nonce. Stream en cryption was done using ChaCha20, and message authentication was the task of Poly1305, and associated data was processed with the help of the addAuthData functionality. This model is a high performance, software-based AEAD solution that is suitable on platforms of which hardware AES acceleration is not available.

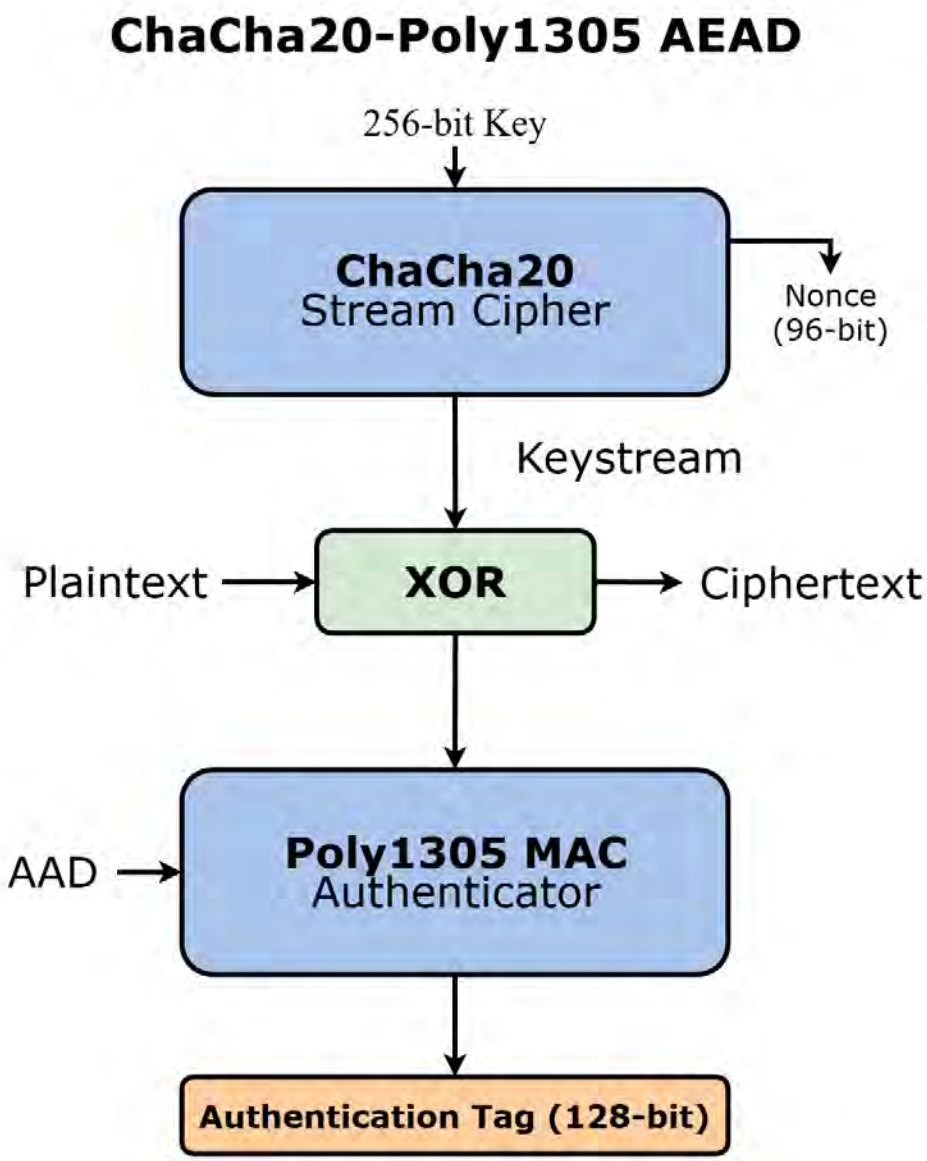


Figure 2.3: Architectural overview of ChaCha10-Poly1305 AEAD encryption workflows

- **TinyJAMBU-128:** TinyJAMBU-128 was written with the help of tinyjambu-128 aead-encrypt function, which follows the NIST Lightweight Cryptography AEAD API. A permutation-based design is used to minimize the amount of software and hardware resources required in the cipher. Its AEAD structure can provide confidentiality and

integrity in a single operation and ciphertext output is enhanced with an authentication tag.

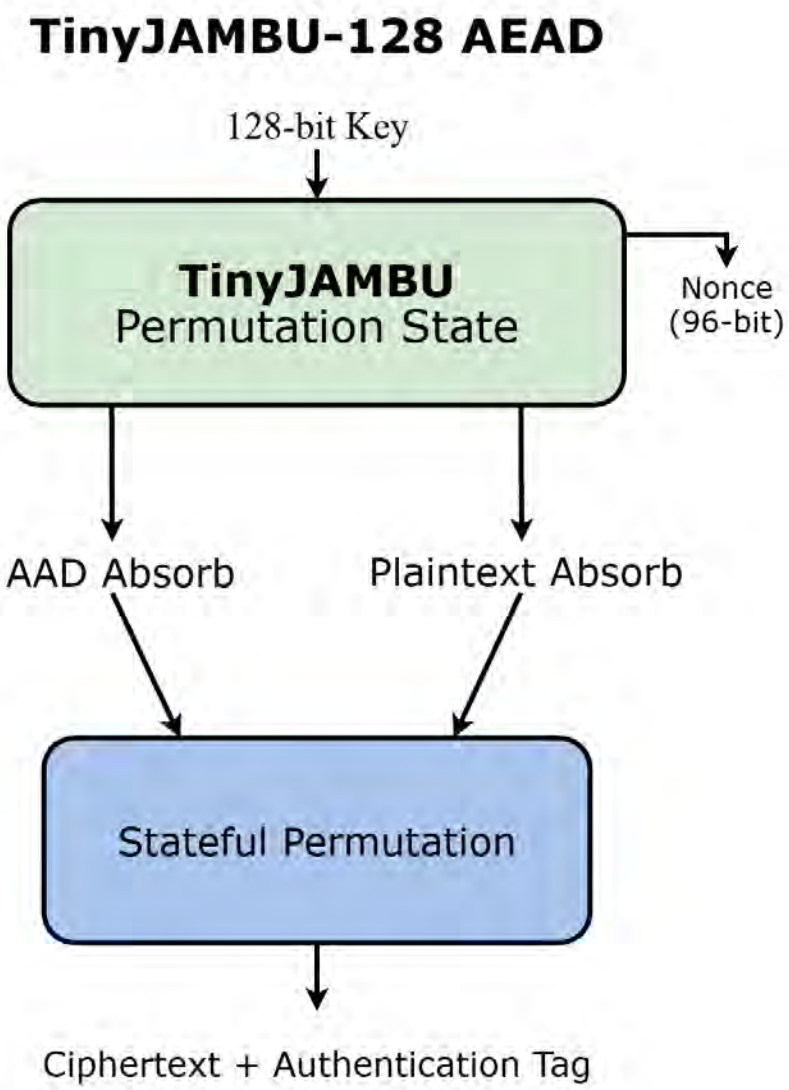


Figure 2.4: Architectural overview of TinyJAMBU-128 AEAD encryption workflows

- **GIFT-COFB:** GIFT-COFB was designed and implemented with the `gift_cofb_aead_encrypt` function, which is a combination of the COFB mode operation and lightweight block ci-

pher GIFT-128. This lightweight AEAD design which is based on a block cipher allows comparative analysis with the permutation-based and stream cipher-based encryption schemes.

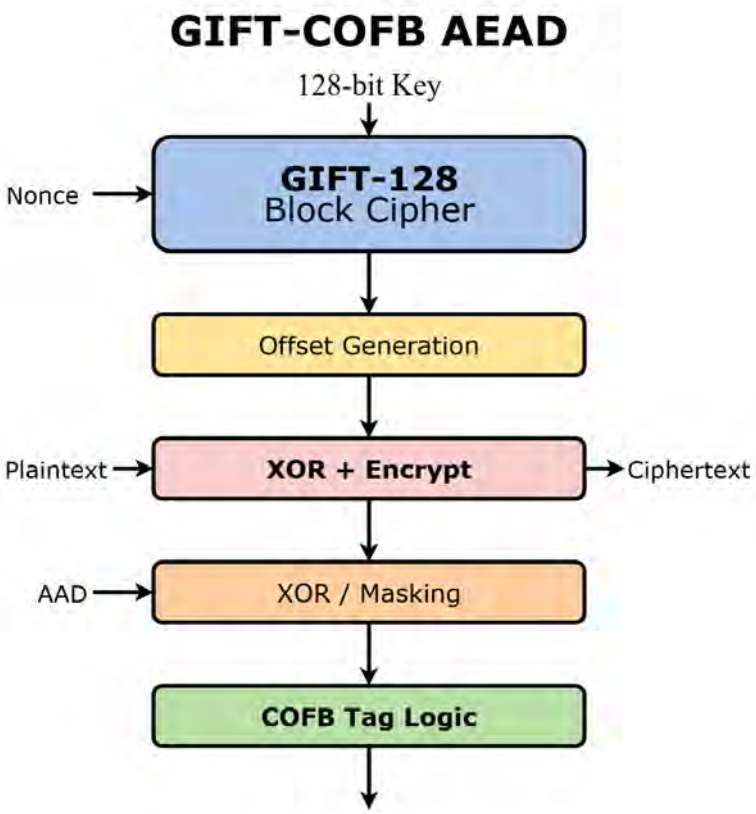


Figure 2.5: Architectural overview of GIFT-COFB AEAD encryption schemes

- **IoT Data Security:**

  In the area of Internet of Things (IoT) application, where large amounts of sensitive information are collected, transmitted, and stored, data security becomes one of the essential aspects of current information systems. The basic principles of data security are the CIA triad, confidentiality, integrity, and availability. The issue of data security is especially critical to Internet of Things (IoT) systems because these systems can often operate in an environment characterized by numerous devices, limited resources,

and an increasing level of cyber threats.

–Confidentiality keeps the data that should not be accessed by any users or devices through restricting access to the data by authorised users or devices. In order to avoid intrusion, any sensitive information in the IoT devices, such as temperature, user, and health-related information should be confidential. Encryption contributes immensely to the provision of confidentiality by ensuring the plaintext is converted into encrypted text and cannot be deciphered without the correct decryption key. In Internet of Things (IoT) systems that operate based on resource-constrained systems such as microcontrollers (ESP32), which have less processing power and memory, demanding lightweight encryption systems such as ASCON are necessary. According to the recent studies [1]. ASCON is an excellent choice in systems such as pharmaceutical cold-chain monitoring that require data integrity and confidentiality since it is highly effective when it comes to secrecy preservation within IoT systems.

–Integrity It is the assurance that data has not been modified, interfered with or corrupted unintentionally or intentionally. This implies that the integrity of data in the IoT systems is quite challenging due to the scattered nature of the IoT devices and the fact that interception or manipulation of the data is possible during transit. To maintain the integrity of data, IoT systems will often employ authentication, which ensures that no one has changed the data. Authenticated encryption systems, such as ASCON AEAD and AES-GCM, provide encryption as well as message authentication that ensures the confidentiality and integrity of the data. Such encryption methods ensure that any modifications done on the encrypted data will be detected safeguarding the integrity of the data being transmitted[1], [2].

–Availability ensures that data is available when required to authorised personnel or devices. Availability is a significant aspect with regard to the IoT systems because any failure or unavailability of essential data, in the case of cold-chain monitoring such as temperature data, can cause significant consequences such as the loss of medications. The IoT systems should be designed to handle large data volumes and resist failures or attacks to ensure high availability. A secure transmission protocol, error correction, and redundancy are common to enhance the availability of data in the IoT systems. Moreover, cloud-based platforms are vital in enhancing the availability of data by maintenance of encrypted information in one repository that users can always have access to. The combination of cloud services and IoT devices allows ensuring the availability of data to analyze and monitor even when local devices fail [7].

One of the main challenges of IoT systems is unauthorized access, as it might lead to the breakage of data, infringement of privacy, and compromise of the system. Authentication and access control measures are necessary components of the IoT data security due to the growing number of connected devices and the risks they expose. Role-based

access can also be used to restrict access to sensitive data in accordance with user roles [6], whereas user authentication can be used to ensure that the system is accessed by authorised individuals or devices. In this study, it is important to prevent attacks by attackers, which involves preventing unauthorised access to the IoT data. This is achieved through the combination of sound encryption methods and protocols such as the ASCON AEAD that ensure that only authenticated devices and users can access and decrypt data.

The IoT system is important to end-to-end security which implies that the data should be safeguarded since its collection at sensors down to the moment it is received and stored in the cloud. To achieve this, the data should be encrypted when it is stored (when it is stored in databases or on cloud platforms) and also when it is being transferred through the network. The information transmitted through the internet is encrypted and protected against man-in-the-middle attacks and eavesdropping by applying secure communication protocols such as the SSL/TLS. Moreover, smaller cryptographic functions such as ChaCha20-Poly1305 and TinyJAMBU are designed to be compatible with very low-processing power devices in IoT, which offer high-security level encryption and authentication without performance and battery life cost [3], [4].

In addition, Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks, which may disrupt the availability of data by overloading devices or networks with traffic, is a relatively common threat to IoT systems. Some of the measures that can be taken in averting such attacks include installation of firewalls, rate-limiting and intrusion detection systems. Also, this work provides another protection aspect because it relies on time-delay decryption protocols that facilitate the challenge in decrypting intercepted data among the attackers. This protocol enhances the general security of the system and prevents the brute-force attacks by introducing a computational delay on the decryption step[1].

The data security issue cannot be overestimated with the steadier growth of IoT systems in such critical fields as healthcare and pharmaceutical logistics. The confidentiality, integrity, and availability of data in the IoT systems must be maintained with the help of strong encryption, effective authentication methods, and secure communication protocols. Lightweight cryptographic algorithms such as ASCON, ChaCha20-Poly1305, and TinyJAMBU are required to design IoT devices so that they can be able to safely transmit and store data with maintaining the resource constraints of embedded systems [2], [1], [7]. To enhance the security of cold chain pharmaceutical products that are sensitive to temperature, this research paper aims at examining different cryptographic methods and their possible implementation as a protection in IoT data.

- **Performance metrics:** like encryption time, throughput, and latency It is crucial to evaluate the encryption in order to know its efficiency. Encryption time is the time

taken to encrypt a certain amount of data. Throughput gives us the information of how much data has been encrypted in a certain period of time and latency refers to the delay in the encryption time. The metrics will be used in this study to determine how efficiently ESP32 can handle the encryption of temperature data without compromising

2.2 Review of Existing Research

Rezvani and Diehl et al. [9] This paper offers one of the first third-party hardware assessments of six NIST LWC Round 2 candidates; SpoC, GIFT-COFB, COMET-AES, COMET-CHAM, Ascon, and Schwaemm/Esch, in order to address the issue of securing IoT devices using lightweight cryptographic algorithms. The authors fully comply with the recently published NIST LWC Hardware API by implementing these ciphers in Verilog/VHDL on Artix-7, Spartan-6, and Cyclone-V FPGAs using a basic-iterative architecture. They then validate their designs in real hardware using power and energy measurements at 40 MHz. According to their results, Ascon had the highest throughput and throughput-to-area ratio, COMET-AES and Ascon were the most energy efficient, Schwaemm/Esch needed the largest area and had the lowest efficiency, while SpoC had the smallest area and the lowest power consumption.In lightweight cipher hardware, the result is a comparative baseline of resource consumption, performance, and energy efficiency that aids in identifying design trade-offs and potential hazards. The study does, however, recognize many limitations, including the fact that results are based solely on a selection of candidates, optimizations are modest (baseline implementations), comparisons between various FPGA types are not exact, and without defined benchmarks, a final ranking cannot be made. All things considered, the study highlights the necessity for uniform evaluation criteria while offering a significant early insight into the hardware viability of NIST LWC options.

koo et al. [10] This survey covers the 32 candidate algorithms that advanced to the second round of NIST's Lightweight Cryptography Standardization, aimed at finding secure and efficient authenticated encryption with associated data (AEAD) schemes for resource-constrained IoT devices such as RFID tags, sensors, and smart cards. It divides the candidates into four categories: permutation-based, stream cipher-based, block cipher-based, and tweakable block cipher-based designs. It finds that permutation-based schemes are the most common, and all of the contenders employ 128-bit keys with different nonce and tag sizes. For efficiency, the majority of algorithms allow parallelism, online processing, and inverse-free operation; nevertheless, 11 contenders, primarily based on block ciphers, failed to achieve the necessary 21122

to the power 112 security criterion. According to hardware evaluation, SAEAES is the least resource-demanding on FPGA and TinyJambu is the most area-efficient (1674 GE); however, 13 candidates did not have hardware performance data, and those that did used inconsistent platforms, making comparisons unfair. In order to facilitate better assessment, the study ends with a request for a standardized hardware evaluation framework. It emphasizes that although there are a number of potential schemes, inconsistent and incomplete data continue to be a limitation.

Madushan et al. [11] In order to address the issue of securing resource-constrained IoT devices where standard cryptographic algorithms are too heavy, the thesis paper "A Review of the NIST Lightweight Cryptography Finalists and Their Fault Analyses" looks at the ten finalist algorithms (Ascon, Elephant, GIFT-COFB, Grain-128AEAD, ISAP, PHOTON-Beetle, Romulus, SPARKLE, TinyJambu, and Xoodyak) in the NIST Lightweight Cryptography (LWC) project. Since these devices are physically accessible and extremely susceptible to side-channel manipulation, the authors categorize the finalists according to their primitives (mostly sponge-based, plus block cipher and stream cipher designs), talk about their structures, parameters, and benefits, and examine known cryptanalyses with an emphasis on fault attacks. They highlight how fault injection techniques like clock glitches, laser glitches, or power glitches can jeopardize security and summarize current attacks like differential, collision, statistical, and ineffective fault attacks applied to some candidates (Grain-128AEAD, PHOTON-Beetle, GIFT-COFB, and Ascon). The study gathers and examines findings from the NIST competition framework and publishes third-party evaluations instead of using any unique experimental hardware configuration. Although many lack a comprehensive study against real-world side-channel and fault attacks, the key conclusion is that none of the finalists have been breached by current assaults and all maintain their stated security levels. NIST did not require resistance to such attacks in its evaluation, which leaves open research gaps for future study. Other limitations include the fact that not all algorithms have been tested equally and that some fault models have not yet been investigated.

Buchanan et al. [2] This study talks about the challenges of applying traditional methods in embedded systems, focusing specifically on the limitations which are due to the resource based constriction in devices like IoT networks. It talks about how lightweight cryptographic methods would be more efficient in these environments, as they use smaller blocks and reduced key sizes and still be able to maintain security efficiently. The paper looks into various lightweight cryptographic methods like PRESENT, CLEFIA etc. It focuses on their performance, how much energy they consume and how effective it is in terms of security in IoT applications. Even though

lightweight cryptography performs better in environments with limited resources, it is still less secure than traditional methods like AES and SHA. However, more research is needed to understand the effectiveness of these protocols. This study focuses on specific cryptographic methods , hence the results could be limited as performance and security varies across different devices and IoT devices. This might also lead to ineffectiveness and vulnerability to specific attack scenarios

Khan et al. [6] This work looks into the security challenges caused by IoT devices and it focuses on how each IoT device would require a specific cryptographic method that would ensure security without compromising efficiency. This study looks into the existing cryptographic protocols, discussing how suitable they would be for IoT environments as different devices have different capabilities. A comparative analysis has been done of several lightweight cryptographic schemes. Evaluating them based on their performance, security and energy efficiency. Furthermore, this paper also looks into the vulnerabilities of existing protocols applied to different devices with different computational capabilities. This paper shows the importance of the need for elastic cryptographics which can adapt to different IoT devices with different computational capabilities and resource constraints. More future research should be done in order to develop more robust solutions with better security and performance. Different IoT devices have different computational capabilities. However, the study assumes that IoT devices use the same protocol which can cause issues in applying a single protocol.

Lee et al. [12] This report presents a lightweight authentication protocol designed to positively upgrade the security and privacy of Internet of Things (IoT) devices where RFID technology has been used. The paper discusses the problems of protecting massive IoT systems that don't have the computing power for typical encryption algorithms and have limited resources. The authors suggest a simple but effective encryption method that uses XOR operations instead of more complicated hash or public-key cryptography to fix this problem. The protocol lets RFID tags and readers authenticate each other, which helps stop problems like cloning and unwanted access. The implementation, confirmed using finite state machine (FSM) modeling and hardware synthesis utilizing Quartus II, shows that this methodology can achieve real-time authentication with minimal power consumption and decreased circuit complexity. The study at last establishes a basis for secure and lightweight encryption in RFID-based IoT networks, achieving a balance between simplicity and proper security against prevalent threats..

Ning et al. [13] This study introduces a hybrid MCDM method of choosing a lightweight cryptographic cipher based on ISO and NIST standards to work with

the Internet of Health Things. Ning recommends a multi-criteria decision-making (MCDM) framework to be used in deciding the most appropriate lightweight cryptographic. Internet of Health Things (IoHT) application cipher in which the devices possess a memory, power, and processing power that are limited. The framework works with CRITIC (Criteria Importance Through Inter-Criteria Correlation) to weight evaluation factors and TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) to rank options.The study uniquely combines ISO/IEC 29192 and NIST security standards to create a standardised way to evaluate hardware, software, and security features like throughput, latency, chip area, power, and key size. The analysis of ten lightweight ciphers, such as PRESENT, LEA, HIGHT, AES, and KLEIN, finds that KLEIN is the best choice because it strikes the best balance between energy efficiency, computational performance, and protection against assaults. The study connects the theoretical and practical sides of choosing a cipher.

Tran et al. [14] This provides a Hardware Architecture of NIST Lightweight Cryptography deployed in IPSec to Secure High-Throughput, Low-Latency IoT Networks. The problem of interest in this study is the hardware implementation of the NIST-approved lightweight crypto graphic algorithm in the IPSec protocol to offer secure communication in high-throughput, low-latency IoT networks. The hardware architecture designed and experimented by the authors is an integration of lightweight ciphers such as ASCON, GIFT, and TinyJAMBU that are the most effective in limited resource embedded systems. The analysis indicates that the implementation and performance evaluation with the use of FPGA brings huge improvements in terms of energy efficiency, encryption speed, and latency in comparison to the conventional IPSec constructions. The architecture offered is compatible with the NIST specifications of lightweight cryptography by ensuring it is compatible with the IoT communication protocols at reduced silicon and power consumption. This text assists in bridging the gap between the design and implementation of cryptographic algorithms in real-life hardware. It demonstrates that by adding lightweight cryptography to the direct IoT communication frameworks, the IoT communication can become secure, efficient, and scalable.

Serrano et al. (2020) [1]: states that because of the performance and side channel limitations of AES-based cipher suites in embedded systems, a hardware based implementation of the ChaCha20-Poly1305 Authenticated Encryption with Associated Data (AEAD) scheme, which is compatible with Transport Layer Security (TLS) 1.3, has been created. The primary objective of the study is to deliver a complete and standards compliant AEAD design that is suitable to the actual implementation of secure communication systems. A full implementation of ChaCha20 and Poly1305

as an AEAD architecture that is fully TLS 1.3 compatible, allowing arbitrary sized plaintexts and other authenticated data (AAD), is among the main strengths of this work. Accumulator and filtering mechanism implementation enhance robustness in real world deployments as authentication issues introduced by fragmented data blocks are effectively resolved. The implementation's relevance to contemporary embedded systems is increased by evaluating it both as a standalone FPGA core and within a RISC-V system-on-a-chip. With 1.4 cycles per byte in standalone hardware and 11.56 cycles per byte in a system level environment, a performance gain of more than 1000 percent performance measure ment reveals notable gains over software implementations. When compared to similar ChaCha20–Poly1305 and AES-based AEAD implementations, the resource utilization is still competitive. However, because more control logic is needed for TLS compli ance, the architecture has a greater area overhead. Furthermore, its appropriateness assessment for ultra-low-power IoT devices is limited by the lack of a thorough power and energy consumption analysis. All things considered, the work makes a significant and useful addition to hardware accelerated cryptography for TLS 1.3.

Banik et al. (2020) [3]: presents the implementation of a hardware accelerated cryptographic system intended to improve secure communication in embedded and resource-constrained situations is examined in this thesis article. The work's main goal is to integrate authenticated encryption algorithms onto specialized hardware in order to get around software-based encryption's performance constraints. The paper demonstrates that hardware acceleration can significantly enhance the performance of encryption and reduce the processing delay and an organized system design can be presented. The criterion of realisticness and system-level analysis that considers cryptographic elements as part of an entire hardware platform, and not as a discrete algorithm, is one of the main strengths of this work. The experimental results are a clear indication that the performance is higher than the software models, proving the suitability of hardware-based security to fast and real-time applications. The study is positioned within previous research thanks to comparisons with relevant implementations and a well organized methodology. The study does, however, have a number of drawbacks. Performance indicators are the primary focus of the study, with little attention paid to energy efficiency, scal ability, and hardware attack resistance all of which are critical for contemporary IoT systems. Furthermore, the conclusions are less broadly applicable because the implementation is limited to a certain hardware platform. Its applicability to new security trends is further limited by the lack of comparisons with more recent lightweight cryptographic standards. All things considered, the study makes a significant addition to hardware cryptography; nonetheless, a more comprehensive assessment would increase its influence

Mihai et al. (2016) [5]: presents the design and construction of a temperature and humidity monitoring system using an Arduino Uno board and a DHT22 sensor. A real-time clock module is also used for time monitoring. The goal of the work is to demonstrate a functioning hardware-software system capable of collecting, processing, and displaying indoor environmental data. The main strengths of the work include its direct and practical system design which is easy to comprehend by the novice and the teacher. The hardware design and wiring schematics, as well as communication protocols, are properly described in the paper. Reproducibility is enhanced by the use of hardware and Arduino development environment. Experimental outcomes confirm the correct functioning of the sensor, and the paper has a comprehensive description of the accuracy of measurement and data formatting. The technical depth of the study is, however, insufficient, because it is mostly descriptive. Only basic functional testing is conducted, without a quantitative analysis of long-term accuracy, reliability, or calibration. Performance factors such as scalability, power consumption, and sampling rate are not discussed. Moreover, the paper's applicability to modern smart monitoring systems is limited because it does not address issues like security, data integrity, or integration with IoT platforms.

Saha et al. (2020) [4]: presents the first independent cryptanalytic evaluation of TinyJAMBU, a lightweight AEAD candidate in NIST's LWC process. The study's methodological contribution of a modified MILP model that overcomes the inadequacies of previous gate counting approaches while staying computationally tractable is its primary strength. This model captures first order correlations between chained AND gates. The authors use this model to show forgery attacks that lower TinyJAMBU's security margin and to derive better differential and linear trails. They demonstrate that full round attacks leave less than an eight bit margin in data complexity and that high probability differentials over 338 of 384 rounds undermine the purported 64 bit authenticity level. By connecting these flaws to TinyJAMBU's low state size, multi block nonce, and tag processing, the work also provides design insights. The analysis has limitations in spite of these contributions. Higher order dependencies are approximated by the improved model, which might ignore correlations. Attacks are theoretical in nature and depend more on idealized assumptions than on actual implementation costs. Other attack routes are not examined in the results, which concentrate on differential and linear cryptanalysis. The study improves knowledge of TinyJAMBU's security overall, although it falls short of a thorough analysis.

Gunathilake et al. (2019) [7]: justifies the need to use lightweight cryptography when deploying Internet of Things devices. IoT devices contain minimal power,

memory and processing capacity. It is becoming considerably popular in the fields such as smart cities, medical equipment and wearable devices, and autonomous vehicles. Consequently, it is increasing the difficulty of maintaining data security on a daily basis. Ensuring the privacy, data integrity and the identity of the user is now a huge challenge. Conventional security systems such as AES and RSA consume a lot of power and computing power. Therefore, they should not be a good option in small IoT devices since it is weighty on the battery and may slow down the performance. The analysis indicates that the lightweight cryptography suits such a device as its consumption is lower and needs fewer hardware but the data does not lose its reasonable level of security. Another network that is discussed in the paper is the LoRaWAN, a low-power and long-range network. LoRaWAN can operate lightweight cryptography to ensure that the device remains secure without consuming a lot of battery. Various lightweight algorithms are also compared in the study. It also examines their size of key, block size and their ability to cope with software and hardware attacks. The paper clarifies that the future IoT systems require the use of lightweight cryptography. The existing solutions are effective, although not flawless. According to the authors, it requires further testing and real-life experiments. Security and performance should also be measured using better standards. In general, the paper demonstrates that lightweight cryptography is important in ensuring that IoT systems are secure and energy efficient.

Latif et al. (2020) [8]: summarizes the security in the IoT and reviews the works on security requirements, key management, and lightweight cryptography. The authors clarify that IoT devices exchange an enormous volume of data amongst themselves. Such machines are extremely dissimilar in nature and capability. It is due to this that they are more prone to cyber attacks. Security has turned out to be a big problem. The paper describes the security of IoT depending on the TCP/IP layers. The weaknesses of each layer vary hence each layer requires a different form of protection. Key management, which is one of the most difficult aspects of IoT security, is also given much attention in the paper. The authors talk about the symmetric and asymmetric key systems. They elaborate that symmetric keys are effective when it comes to communication between devices since they are quick, and consume a small amount of energy. These keys are however hard to share in a secure way. The authors emphasize on asymmetric cryptography in authentication. Elliptic curve cryptography is appropriate since it is based on smaller keys and it needs less processing power compared to the older systems of public key cryptography. The authors examined various lightweight cryptographic algorithms, with a comparison of the speed, energy consumption and security strength. The paper concluded that a mixed/hybrid approach is the most suitable to use with the systems of the IoT. Symmetric encryption with a lightweight kind of encryption should be aimed at securing inter-device data. Lightweight tech-

niques based on ECC must be employed when a gadget communicates with a server. The paper demonstrates that lightweight cryptography should be used alongside adequate key management to achieve secure yet scalable and energy-efficient IoT systems.

Moura (2024) [15]: focuses on how to make encryption faster and more efficient for low-powered IoT systems. Such devices can handle sensitive data but have very limited power, memory, and processing ability. Traditional encryption methods require too many resources, as a result, making them unsuitable for such devices. Hence,to solve this problem, the paper focuses on lightweight cryptography and hardware acceleration as practical solutions. It examines three authenticated encryption algorithms. Ascon, AES-128 in CCM mode, and ChaCha20-Poly1305. The test is done on a low-complexity RISC-V processor called Ibex. The author compares how these algorithms perform when they run only in software and when they are accelerated using special processor instructions and dedicated hardware blocks. Special instruction set extensions like Zbkb, Zkne, and XAscon are added to speed up cryptographic operations. It shows that hardware acceleration greatly improves performance and energy efficiency, especially for Ascon and AES-128. AES-128 shows the highest performance gain, with Ascon in second, while ChaCha20-Poly1305 benefits less due to its design being more software friendly. A small increase in chip area significantly improves energy efficiency. As a result, better speed and lower power use without significant hardware cost. The paper shows that hardware acceleration is very effective for devices that encrypt a large number of messages. Ascon performs well when combined with RISC-V extensions. Overall, the study proves that combining lightweight cryptography with hardware acceleration is a strong solution for secure, energy-efficient, and resource-limited embedded systems.

Khedkar et al. (2016) [16]: explores ASCON's suitability across a wide range of authenticated encryption applications such as RFID tags, wireless sensor nodes, embedded systems, and high-performance computing. The authors implemented multiple Ascon hardware variants which were optimized for various goals such as minimal area, low power, low energy, and maximum throughput, showing that Ascon can achieve extremely small area footprints (as low as 2.6 kGE) as well as very high throughput (over 5 Gbit/s) with modest hardware cost. The paper shows that Ascon can be efficiently protected against first-order differential power analysis using threshold implementations, with significantly lower overhead compared to AES-based designs. From practical side-channel evaluations, we can see that the protected implementations successfully resist power analysis attacks. The results show that Ascon offers exceptional flexibility, strong hardware efficiency, and practical resistance to side-channel attacks, making it well-suited for a broad range of real-world authenticated encryption.

2.3 Summary of Key Findings

Table 2.1: Summary of the literature review

Papers	Perf. analysis	Crypto analysis	Impl./ HW	Auth.	Energy / Power
Rezvani et al. [9]	✓	✓	✓	✗	✓
Koo et al. [10]	✓	✗	✓	✗	✓
Madushan et al. [11]	✓	✗	✗	✗	✗
Buchanan et al. [2]	✓	✗	✗	✗	✓
Khan et al. [6]	✓	✗	✗	✓	✓
Lee et al. [12]	✓	✗	✓	✓	✓
Ning et al. [13]	✓	✓	✓	✗	✓
Tran et al. [14]	✓	✓	✓	✗	✓
Gunathilake et al. [7]	✓	✓	✓	✓	✓
Latif et al. [8]	✓	✓	✗	✓	✓
Moura [15]	✓	✗	✓	✓	✓
Khedkar et al. [16]	✓	✗	✓	✓	✓
Serrano et al. [1]	✓	✗	✓	✓	✗
Banik et al. [3]	✓	✓	✓	✓	✓
Bogdan [5]	✗	✗	✓	✗	✗
Saha et al. [4]	✗	✓	✗	✓	✗

Table Description

The table shows a comprehensive summary of papers which focused on lightweight cryptography, particularly in domains such as Internet of Things, their security low powered devices, authenticated encryption and implementations. This table evaluated the papers in five key feature columns, Performance Analysis (Perf. analysis), Cryptographic Analysis (Crypto analysis), Implementation/Hardware (Impl./HW), Authentication (Auth.), and Energy/Power. A ✓ symbol represents the presence of that column name feature in the paper while "x" represents its absence. Through this binary classification, we can ensure a structured comparison between the papers, looking into trends such as emphasis on performance and implementation in hardware focused papers, on the other hand, review type papers focus on cryptographic analysis instead of energy usage concerns. Overall

it reflects a mix of theoretical cryptanalysis, real life implementations and application based studies.

Column Wise Description from Table

Perf. analysis

The "Performance Analysis" column, written in short as "Perf. analysis" tells us whether the given paper evaluated or tested the performance aspects of the cryptographic algorithm. This analysis might be done through several categories such as data processing speed, latency, resource utilization, power consumption and so on. This column helps us assess if the given research paper looked into practical efficiency by testing performance metrics which is essential when it comes to real life implementation or deployment of a system.

Crypto analysis

The "Crypto analysis" column, which stands for "Cryptographic Analysis", checks whether the given paper examined or did in depth cryptanalysis of cryptographic algorithms. This may include several testing such as linear cryptanalysis, vulnerability assessment, proof of security or resistance to attacks and so on. This testing is very essential as it helps us understand the robustness or effectiveness of the discussed cryptography, specially in security focused research.

Impl./ HW

The "Impl./ HW" which stands for "Implementation / Hardware, tells us whether the given paper discusses or presents actual implementation, especially focusing on hardware aspects. This may include description of prototypes, hardware designs, or code execution for cryptography algorithms. This column is essential as it highlights how theory concepts can be deployed into real life devices.

Auth.

"Auth", which stands for "Authentication" , checks if the paper looked into authentication mechanisms or features in terms of cryptography. This may include topics like Authenticated Encryption, message authentication codes, verifying integrity, or systems ensuring data authenticity. This column is very important in terms of analyzing security as proper authentication helps prevent tampering or spoofing.

Energy/Power

This column examines whether the given paper discussed energy consumption or power efficiency of the cryptographic implementation or the devices. The discussion might include analyzing battery life impact, power draw, energy required for operation and optimizing the energy level for low powered systems. Considering energy or power efficiency is important as minimizing power usage is required for using it in resource constrained settings.

Paper Wise Description from Table

Rezvani et al.

Rezvani et al. evaluate hardware implementations of six NIST LWC Round 2 candidates like Ascon and GIFT-COFB on FPGAs. From the table we can see that this paper reflected on performance analysis, cryptographic analysis, hardware implementation and energy/power, but didn't quite focus on the authentication part. The paper evaluated hardware implementations of six NIST LWC Round 2 candidates on FPGAs, for performance analysis, cryptographic analysis, implementation/hardware, and energy/power, but not for authentication. Performance analysis includes TPA ratios, Ascon at 0.790 Mbps/LUT, throughput up to 2583.3 Mbps for Ascon, and comparisons shows SpoC's smallest area but Ascon's highest TPA. Cryptographic analysis covers security parameters like 128-bit keys and states, with brief resistance notes but no new attacks. Implementation/hardware uses Artix-7, Spartan-6, and Cyclone-V FPGAs with LWC HW API, RTL designs, and Minerva optimization, e.g., SpoC at 1172 LUTs and 154.5 Mbps. Authentication is implied in AEAD (x standalone). Energy/power (/) measures average energy per bit at 40 MHz, emphasizing lightweight targets.

Koo et al. [10]

Koo et al introduces a family of block ciphers known as CHAM, which are built using simple arx operations, focusing on performance analysis, implementation/hardware, and energy/power. Performance analysis shows bit-serial at 5.0 Mbps/100KHz and round-based at 80.0 Mbps, with comparisons to SIMON/SPECK highlighting smaller areas (665 GE for CHAM-64/128). Implementation/hardware details Verilog synthesis in UMC90/180 and IBM130, with architectures saving 259 GE via stateless key schedules. Energy/power is optimized for low-power IoT, with smaller sequential logic reducing consumption.

Madushan et al. [11]

Madushan et al. review NIST LWC finalists, focusing on performance analysis and cryptographic analysis only. Performance analysis discusses trade-offs like smaller blocks/keys

for efficiency, with metrics such as cycles/byte. Cryptographic analysis summarizes attacks like DFA on Ascon at 2^{96} complexity and SIFA on SPARKLE. The paper explains trade Madushan et al. provide a clear and easy review of the NIST lightweight cryptography finalists, making the paper a useful reference for IoT security. The work mainly covers performance and cryptographic analysis, but not hardware implementation, authentication details, and energy use are not explored in depth. For performance, the authors explain common design trade-offs, such as using smaller keys or fewer rounds to improve speed and reduce cost. They describe how designs like Ascon's sponge construction or TinyJAMBU's compact NLFSR aim for efficiency, without focusing on exact throughput numbers. The cryptographic analysis is the strongest part of the paper. It classifies finalists by their underlying primitives and summarizes known fault attacks, including differential fault analysis on Ascon with 2^{96} complexity and fault attacks on SPARKLE. The paper also compares strengths and weaknesses, such as Ascon's side-channel resistance versus Romulus's larger internal state.

Buchanan et al. [2]

Buchanan et al. consider lightweight cryptography through a highly realistic perspective on small device limits. The article is concerned with performance and energy consumption, so the in-depth theory of security, hardware architecture, and authentication issues are not covered in the article. Performance-wise, the authors describe the performance loss due to heavy algorithms such as AES, and the lightweight counterparts that are more balanced in terms of speed and low overhead. The best section of the paper is energy and power that demonstrates why lightweight cryptography is important to battery-powered equipment such as sensors and RFID, and in a wireless network, power consumption is a factual issue.

Khan et al. [6]

Khan et al. survey lightweight security protocols of IoT systems that possess very different capability of devices. The paper is devoted to performance, authentication and energy efficiency, rather than to cryptographic analysis or hardware design. The concept of performance is addressed at the levels of the IoT, and demonstrates how delays and inefficiencies may be introduced when protocols are not flexible. The theme of authentication is also an important concept, where the authors suggest the use of adaptive or elastic protocols that adapt to weak and strong nodes. It focuses on power and energy consumption, particularly the mismatch of small devices to servers that are always powered up and insists on designs to defend against rapid battery depletion of small devices.

Lee et al. [12]

The authentication protocol suggested by Lee et al. is very simple and is used in IoT and RFID systems with the help of XOR operation primarily. The analysis includes performance, hardware, authentication, and power consumption, but does not include cryptographic analysis. Performance claims are concerned with low computation cost of tags, readers, and databases with internet communications. Hardware viability is depicted with the help of Verilog and QuartusII to create net-lists. Authentication is the main goal, aiming to prevent counterfeiting and unauthorized access with minimal complexity. Energy efficiency is also central, as the lightweight design suits battery-limited RFID tags that need long operating life

Ning et al. [13]

Ning et al. propose a decision-making framework to choose the best lightweight cipher for Internet of Health Things systems. Their approach evaluates performance, security, hardware cost, authentication support, and energy use together. Performance and hardware factors include throughput, block size, and gate count, showing how some ciphers fit better in small medical devices. Cryptographic analysis follows NIST and ISO standards when it is comparing resistance to attacks and key management strength. Authentication is treated as an way for protecting medical data from tampering. Energy and power are heavily weighted, since many IoHT devices are battery-powered. Based on all criteria, the study ranks PRESENT-80 as the best overall choice for healthcare IoT.

Tran et al. [14]

Tran et al. propose a hardware design of Ascon that is built directly into IPSec for fast and low-latency IoT networks. Their work mainly covers performance, cryptographic analysis, hardware implementation, and power aspects, while authentication itself is not discussed separately. The performance analysis indicates that speed is very high with minimal delay. The design can go to 8.806 Gbps at 104 MHz and is capable of computing 128 bits per clock cycle. It also attains 427 ns latency with 2812 FPGA slices. In comparison to the previous designs, this method provides approximately 57 percent greater throughput and the primary objective is not to go as fast as possible, but to decrease latency. The cryptographic discussion justifies the reason why Ascon is appropriate in IPSec. It also emphasizes security of the AEAD design of Ascon sponge with 320-bit internal state and complies with the NIST guidelines of lightweight cryptography, but nothing novel is presented to prove security. One of the key elements of the paper is hardware implementation. The authors implement the Xilinx Virtex-7 FPGA having an AXI Stream interface to provide a broad data path. Hardware savings is done through an iterative loop and an unrolled eight-round permutation is implemented to process

associated data and plaintext. This architecture allows very high peak throughput and eliminates delays associated with changing between crypto cores when using IPSec. The issue of energy and power is taken within the perspective of an IoT gateway. The architecture is based on low power and low latency of real-time traffic. An overall performance, power, and area performance is presented to demonstrate efficiency gains, although the actual values of energy per byte are not presented. In general, the article indicates that Ascon might be easily incorporated into the high-speed and low-latency IPSec systems to be used in the IoT.

Gunathilake et al. [7]

Gunathilake et al. explores the concept of lightweight cryptography as it relates to the next-generation smart IoT devices and pays significant attention to the real implementation of the concept in the LoRaWAN networks. They do all the major aspects in the table such as performance, security, hardware, authentication and energy use. In the context of performance, the paper analyzes quality-of-service requirements like low delay, high efficiency, and superb throughput in the contemporary 5G-based networks. It demonstrates that LoRaWAN facilitates long-range communications and the range of the communication can be up to 20 to 25 kilometers and the power consumption remains very minimal. According to the authors, their design is able to reduce the power consumption related to encryption by approximately 26.2 percent, decrease the overhead at end systems by 75 percent, and decrease the overhead at network servers by 48 percent. The cryptographic analysis looks at such algorithms as AES, RSA, and DES, and the manner in which they ensure confidentiality, integrity, and authenticity in an IoT system. Such usual threats that are discussed in the paper include known-key attacks, replay attacks, and eavesdropping, and proposes some light cryptographic enhancements to overcome these threats. Implemen tation and hardware issues are also addressed in depth and the limiting chip size, cost, memory and power constraints in particular. The authors emphasize on LoRaWAN-based design that require extremely small memory, low data rates and minimal on board storage, which is applicable to devices such as RFID tags and simple sensors. Authentication is handled as a significant need to safeguard the privacy of the user and create confidence in smart environments, such as smart cities, smart healthcare, and smart military. Energy and power efficiency are also subjects of the study. Focusing on the low-power, wide-area networking, battery-powered and green networking concepts. The proposed lightweight solutions across sensors, edge devices, and cloud layers demonstrate apparent energy savings and enhanced efficiency and can therefore be used in resource-constrained IoT deployments.

Latif et al. [8]

Latif et al. examine the management of keys and the lightweight cryptography application in IoT systems. They classify security requirements according to the various TCP /IP levels, and this assists in demonstrating where each type of protection is required. Performance, security, authentication, and energy use are the areas of interest of the paper, although not hardware designs. In the discussion of performance, the authors describe the trade-offs that are typically involved in speed, efficiency and power. In particular, they propose on-the-fly key generation in small devices and ECC in case of communication between the device and servers since it provides high security even when the keys are small. They also contrast various algorithms based on block size, key size and round complexity, stating where each algorithm would be best. Lightweight cryptographic algorithms are also discussed in the paper, including DES, AES, ECC, and lightweight ones, including PRESENT, HIGHT, and TEA. They pay attention to what these algorithms guard, such as data privacy, data integrity, and identity, and what can be vulnerable to them on physical, network, and application layers. The issue of authentication is discussed as one of the primary concerns across the paper. The authors propose to use simple symmetric ciphers in cases where devices communicate with each other, and ECC-based methods when devices communicate with servers. It is emphasized that proper exchange of keys at various layers is necessary to avoid the frequent attacks.. The paper continues to revisit the importance of low-energy encryption in sensors and smart devices, and outlines the way lightweight cryptography can reduce computation and energy consumption, which is essential to the battery-powered IoT systems.

Moura [15]

Moura's thesis focus is to accomplish higher speeds and energy efficiency of AEAD algorithms on smaller embedded systems. It is aimed at such algorithms as Ascon, AES-128 CCM, and ChaCha20-Poly1305 and accelerates them with custom RISC-V extensions. Due to this the paper is very much comprehensive on the topics of performance, hardware design, authentication, and energy use, but lacks deep coverage of cryptographic theory. The performance is highly transparent in terms of performance. The accelerated designs enhance the performance by approximately 95 percent of AES-128 CCM, 60 percent of ChaCha20-Poly1305, and approximately 5 percent of Ascon. It is compared in detail in terms of clock cycles and area. To illustrate, AES-128 CCMdrops on software platforms can take more than 2.1 million cycles whereas with hardware support, AES-128 CCMdrops take approximately 107 thousand cycles. Ascon works particularly well in IPv6 context, being completed after approximately 50 thousand cycles and thus being one of the most notable among the algorithms tried. There are no new security proofs and attacks introduced in the work. Discussion on cryptography remains scarce and largely

describes the reasons why one can trust such algorithms, including the choice of Ascon by NIST and its known ability to withstand typical problems, including side-channel attacks. The thesis is based on hardware implementation. The author works with RISC-V Ibex core with her own extensions, which are Zbkb and Ascon dedicated. It is synthesized using designs using 90 nm UMC technology, and has been reported to make designs as large as approximately 34 thousand NAND2 gates to AES-128 CCM. An Ascon core is also integrated as a hardware-only, also on a custom protocol. Authentication is done with AEAD offering data confidentiality and integrity on low complexity processors. One of the most effective outcomes is energy and power saving. As an example, the AES-128 CCM gas consumption has decreased by approximately 12,808 nJ used in software and only 746 nJ in hardware. These savings are primarily in the number of instructions and considerably reduced clock cycles and the approach is highly appropriate to low-power embedded systems.

Khedkar et al. [16]

The authors build several hardware versions of Ascon, each made for a clear goal such as very small size, low power use, low energy use, or very high speed. Their results show that Ascon can be extremely compact, down to about 2.6 kGE, and also very fast, going beyond 5 Gbit/s without heavy hardware cost. The paper also shows that Ascon can be protected well against first-order power analysis attacks using threshold techniques, and this needs much less extra hardware than similar AES designs. Real attack tests confirm that these protected versions can resist power-based attacks. Overall, the study proves that Ascon is flexible, hardware efficient, and secure in practice, which makes it a strong choice for many authenticated encryption applications in the real world.

Serrano et al. [1]

ChaCha20-Poly1305 implementation to TLS 1.3 is introduced by Serrano et al. with the introduced fault detection. It concentrates on performance, hardware design and authentication, but not cryptoanalysis and energy measurements. The idea is to have a speedy and secure alternative to designs based on AES, particularly with enhanced side-channel attack resistance. The design is very efficient with regard to performance. The single core has a performance of approximately 1.4 cycles/byte, and 11.56 cycles/byte when embedded into a RISC-V platform. It can have a maximum clock speed of 166 MHz resulting in a throughput of nearly 0.95 Gbps. Although AES-GCM is capable of a higher raw throughput, as the authors note, their design provides a significantly higher level of side-channel protection and is approximately 15 times faster than pure software implementation. Implementation Hardware Implementation is comprehensive. It is designed on Virtex-7 FPGA and it consumes approximately 10,808 LUTs and 3,731

flip-flops. It also incorporates several quarter-round modules and multiply-accumulate modules, integration via a TileLink bus as well as a fault filter to work with fragmented data blocks. The overhead is also examined in the study and overhead in the case of bus communication can be attributed to 70 percent of overhead in the system in comparison to standalone operation. Authentication is offered by using the Poly1305 message authentication code that will validate information integrity across the end-to-end communication in TLS 1.3. Nevertheless, the small area and high throughput imply good efficiency, despite the fact that no reported numbers of energy per byte or power are presented.

Banik et al. [3]

Banik et al. suggest a lightweight AEAD, GIFT-COFB, which is a combination of GIFT block cipher and COFB mode. The coverage presented in the paper is good in all key areas and this is why all features are indicated as present in the table. In terms of performance, the authors have very good scores in constrained environments. Hardware synthesis demonstrates a bandwidth of approximately 3271 GE, a throughput of 133.3 Mbps at 10 MHz, and low power consumption, e.g. 2.09 nJ of 16 bytes of related data or plaintext. A number of design modifications are addressed in order to enhance efficiency as well as security. The cryptographic analysis comprises formal security proofs, lower bounds as well as resistance to differential and linear attacks. COFB feedback matrix was also modified to simplify hardware design and alter the linear layer of GIFT to enhance software performance. To be implemented, the paper provides round-based hardware design on UMC 90 nm, UMC 180 nm, and IBM 130 nm technology, and comparing both shared and unshared designs. Through the unshared COFB design, very small hardware sizes are obtained, e.g. 3446 GE. Authentication is another fundamental aspect of the scheme, in that GIFT-COFB has confidentiality as well as authenticity and provable security, including the proper processing of empty inputs. Energy and power consumption are also under close consideration and indicate a low power consumption of about 118.8 uW and a high energy efficiency particularly on longer messages and this makes the scheme a good fit in the IoT and other low-power consuming devices.

Bogdan [5]

The paper by Mihai et al. explains a simple operational configuration that will be used to measure temperature and humidity with the help of a DHT22 sensor and an Arduino Uno board. The article is more on ensuring that the hardware and the software coexist without any hitches, rather than speed or high-level analysis. It have a more implementation and hardware oriented focus. The authors do not quantify such performance values as throughput or delay. Rather, they stress on pragmatic advantages like low

price, non-technical simplicity of installation, and that the system is free and can run on any operating system. Security does not come into the picture either. Encryption, authentication and any form of cryptographic process is not applied in the system, as it is intended to perform simple environmental monitoring. The construction and usage of the system is described in most parts of the paper. It outlines the Arduino Uno and its ATMEGA328 microcontroller, the manner in which the DHT22 sensor can be communicated with via a one-wire protocol and other components such as a 5V power supply, battery, real-time clock, and LCD display are attached. The program is coded and uploaded with the help of Arduino IDE. Practically, the device is capable of measuring temperature to an approximation of 0.5degC and humidity to an approximation of 2 percent. The use of power and energy is not optimized and measured, and focus remains on ease of use, learning, and simplicity and not on secure or energy-efficient IoT design.

Saha et al. [4]

present a deep third-party security study of TinyJAMBU. Their work mainly focuses on cryptographic analysis and authentication, but not speed, hardware design, or power usage. The paper does not discuss performance results such as throughput, latency, or implementation efficiency. Instead, it concentrates on how likely certain attacks are, for example showing a 338-round differential attack with a probability around $2^{-52.65}$. The study introduces a refined MILP model which better captures correlations between AND gates. Using this improved method, they find a 338-round differential trail with 76 active AND gates, where 12 pairs are correlated. When multiple trails are evaluated, the effective probability drops to about $2^{-62.68}$. They also identify a full 384-round differential trail with probability $2^{-70.64}$. These results expose structural weaknesses in TinyJAMBU's NLFSR-based keyed permutation. Authentication is examined through forgery attacks. The paper shows a nonce-respecting forgery attack that can be repeated using only three nonce blocks, which breaks the claimed 64-bit security at 338 rounds. Energy and power aspects are not discussed at all, as the goal of the paper is to evaluate theoretical security margins. The authors mention that only about 12 percent of the rounds remain unattacked and that the gap to full-round attacks is lower than 8 bits of data complexity, meaning a tighter security margin than expected

Chapter 3

Requirements, Impacts, and Constraints

3.1 Final Specifications and Requirements

The hardware and software in this project were appropriate to an Internet of Things (IoT) setting with limited computing capabilities. The main hardware materials will be an ESP32 DevKitC and a DHT22 temperature sensor. ESP32 was chosen because it has low power consumption, inbuilt Wi-Fi and enough computing power to handle real-time data encryption and secure transmission.

The Arduino Integrated Development Environment (IDE) was used in developing the system. A number of lightweight Authenticated Encryption with Associated Data (AEAD) algorithms were introduced to test how well they could fit on a constrained IoT device, such as ASCON-128 AEAD, TinyJAMBU, and GIFT-COFB. Moreover, the ChaCha20-Poly1305 AEAD cipher implementation, which is software-based, was introduced to compare it. The ESP32 hardware cryptographic accelerator was used to implement AES-128 GCM to provide a high-performance reference implementation.

The mitigation of replay attacks was done using a nonce based encryption scheme with a monotonic counter. A windows based host system was used on data communication, measure execution time, and performance analysis. The system design as a whole follows stringent limits on the use of memory, processing power and energy to make sure the system can work in a resource-constrained IoT environment.

3.2 Societal Impact

The study increases the security of the pharmaceutical cold-chain monitoring systems by avoiding the unauthorized access and manipulation of the temperature data. The ability to have safe and precise temperature monitoring will help directly in supporting

public health by making sure that medicines and vaccines do not spoil during storage and transportation. The study will help in the choice of effective and viable security solutions by low-cost IoT devices through the evaluation of various lightweight cryptographic technologies, such asASCON, Tiny JAMBU, andGIFT-COFB. This has enhanced the access and stability of secure healthcare monitoring systems especially in resource limited settings.

3.3 Environmental Impact

Lightweight cryptographic tools can be used to enhance the energy efficiency of a system by lowering the amount of computational resources and energy used. Specially developed algorithms like TinyJAMBU and GIFT-COFB allow decreasing the energy consumption and increasing the life of devices when operating in constrained settings. Temperature monitoring, which is secure, also minimizes wastage of the pharmaceutical due to unfavorable storage conditions. The system indirectly lowers the impact of pharmaceutical products on the environment by preventing the spoilage of medications thus decreasing the production, transportation, and disposal of pharmaceutical products.

3.4 Ethical Issues

The main ethical duty of this research is to ensure that the temperature monitoring data has a high degree of confidentiality and integrity. Loss of unauthorized access to or manipulation of pharma ceutical storage data may lead to severe health consequences. Authenticated encryption algorithms like ASCON, ChaCha20-Poly1305, TinyJAMBU, and GIFT-COFB are used to prevent the alteration of data as well as guarantee reliable communication. All data that is transmitted is encrypted in order to maintain privacy and data relating to the identity are utilized to verify device identity. The system architecture has a balance between high security and usability, such that it does not impose irrational controls to the authorized users.

3.5 Standards Compliance

The primary authenticated encryption algorithm that was tested in this research is called ASCON, and it is a standard that is in accordance with the NIST Lightweight Cryptography (LWC) standard. Other lightweight algorithms, such as TinyJAMBU and GIFT-COFB, are evaluated to give a more comprehensive comparative study. GECM AES-128 with hardware acceleration is based on the established cryptographic standards, and ChaCha20-Poly1305 is a popular AEAD standard that runs on software. With proper

nonce management and the employment of AEAD schemes, the system aligns with the basic security standards of the core IoT, such as confidentiality, integrity, authentication, freshness and replay protection.

3.6 Project Management Plan

The project was implemented in several stages that included system design, implementation, testing, performance analysis and review of literature. The first steps were dedicated to the knowledge of lightweight cryptographic methods and IoT security issues. A performance analysis was carried out after the implementation of ASCN, ChaCha20-Poly1305, TinyJAMBU, GIFT-COFB and AES-128 GCM.

Time and resources were controlled with the help of open-source tools, available cryptography libraries, and cheap hardware parts. Frequent monitoring of progress was to make sure that it was done within the academic schedule..

3.7 Risk Management

The risks such as the processing power being modest of the ESP32, difficulties in using a variety of multiple cryptographic libraries, and uncertainty in performance measurement are possible. This was done by selection of lightweight algorithms, optimization of the codes and application of a uniform test environment in order to mitigate these risks. None devices that allow manipulation of data and replay attacks were mitigated by using nonce-based encryption and authenticated data. The usability was also not compromised by overlooking the performance of other security mechanisms.

3.8 Economic Analysis

The system is cost effective since the components used in the system like the ESP32 microcontroller and the DHT22 sensor are of low cost. Lightweight cryptographic algorithms minimize long term operational expenditure as there is a minimization of the amount of energy used. The solution helps to save costs throughout the supply chain by avoiding losses incurred by pharmaceutical companies due to any breach of temperature during product transportation or data alterations. It is cost effective in implementation and supports various lightweight encryption algorithms hence it can be deployed in large scale in practical use.

Chapter 4

Methodology

4.1 Design Process or Methodology Overview

To understand the efficiency of the cryptography primitives on a small device and at the same time ensure the security of two-way sensor data with authorized encryption and decryption, the research methodology was a comparative, empirical design. Besides deploying a secure communication protocol that delivers secrecy, integrity, authentication, freshness and replay protection, the process was divided into three steps namely; data preparation, system installation and performance evaluation.

Phase 1: Data Preparation

The data of the Medicine Quality Dataset in the form of simulated sensor readings was analyzed to ensure that the input information was valid, clear, and ready to be used as the encryption payload. In order to be able to isolate the one important feature (Storage Temperature) that is used in the real time encryption and decryption experiments, synthetic data had to be generated, missing values were to be imputed (Section 4.3.2), and data had to be reduced (Section 4.3.5). These generated temperature values, which are readings that resemble periodic sensor readings, are encrypted, transmitted, stored and decrypted by the ESP32 via an AEAD based security protocol.

Phase 2: System Implementation

Phase 2 is organized into three conceptual components to explicitly segregate cryptographic functionality, hardware realization, and protocol-level security design.

4.1.1 Encryption and Decryption Implementation

At this point, the ESP32 DevKitC was equipped with five authenticated encryption with associated data (AEAD) ciphers: ASCON-128, AES-128 GCM, ChaCha20-Poly1305, TinyJAMBU-128, and GIFT-COFB. While preserving a uniform security interface, each cipher was integrated using its corresponding standard or reference library.

For ASCON, the functions `ascon_aead128_encrypt`, `ascon_aead128a_encrypt`, `ascon_`

`aead128_decrypt`, and `ascon_aead128a_decrypt` and `ascon_aead128a_decrypt` were used to provide authenticated encryption and verified decryption in compliance with the NIST Lightweight Cryptography standard. As a performance benchmark, AES-128 GCM was developed utilizing the hardware accelerator of the ESP32 using the mbedTLS library. ChaCha20-Poly1305 was realized via the Arduino Crypto library utilizing the ChaChaPoly AEAD interface. Using their respective NIST-compliant AEAD reference functions, TinyJAMBU-128 and GIFT-COFB were built, generating ciphertext and authentication tags in a single operation.
To make sure that timing results solely represented cipher computation overhead and excluded sensor access, communication, and data serialization delays, encryption and decryption latency was only assessed for cryptographic operations utilizing the `micros()` function.

4.1.2 Prototype Development

An ESP32 DevKitC microcontroller interfaced with a DHT22 temperature and humidity sensor was used to create the hardware prototype. The Arduino IDE was used to configure the ESP32 to periodically gather real-time temperature data, format it into a lightweight payload, then encrypt it using AEAD before sending it.
Encrypted sensor data can be sent to a host system or cloud platform for storage and analysis thanks to the prototype's Wi-Fi connectivity. Encrypted payloads were received, test execution was coordinated, and timing measurements were recorded using a Windows-based host environment. Fair and repeatable performance comparisons are ensured by the modular software design, which permits switching between different cryptographic methods without changing the sensor or communication components.

4.1.3 Secure Communication Protocol Design

A lightweight security protocol was devised to ensure freshness, replay protection, and authenticated data integrity. A device specific static salt and a monotonically growing counter are used to create a unique nonce for each encryption operation.

The counter is stored in non-volatile memory to prevent any reuse of nonce during power transitions. Associated Authenticated Data (AAD) including the device identifier, protocol version, and sequence information is included in the AEAD procedure. This will ensure that there is authentication of significant data without encrypting it, which will avoid manipulation. Access control and message authenticity are guaranteed through rejecting payloads with invalid authentication tags, nonce values, and AAD during decryption. This protocol has been designed to ensure that the maximum amount of computation overhead is kept to the minimum without compromising on the security strength. It provides confidentiality, integrity, authentication, freshness and replay protection. Firstly, both the device (ESP32) and the server know a 128-bit master key

K_m but the device maintains a persistent counter and the final nonce is maintained on the server. Once it has been initialized, the device reads a value of DHT22 temperature sensor and its current time and generates a pay load $P = (temp, ts)$. Once the counter is increased by the device, it builds a nonce$N = (ID_D \parallel ctr_D)$, which is unique. With this nonce, a per-message session key is derived as$K_s = H(K_m \parallel N)$. The plaintext is then encrypted and authenticated using an AEAD cipher (ASCON in our case), with the device identity as associated authenticated data (AAD), producing a ciphertext and authentication tag. This transmitted message from the device to the server consists of (ID_D, N, C, T) . After transmission, the nonce freshness is verified by the server to prevent replay attacks, then it derives the session key to perform authenticated decryption. If the integrity verification is successful, the plaintext is stored and the last nonce is updated by the server

Initialization
D, S share a 128-bit master key K_m
D stores a persistent counter $ctr_D = 0$
S stores `last_nonce[ID_D]` $= 0$

Sender Side (ESP32 / Device D)
D $\rightarrow$ $temp =$ `ReadTemperatureSensor`()
D $\rightarrow$ $ts =$ `GetCurrentTimestamp`()
D $\rightarrow$ $P = (temp \parallel ts)$
D $\rightarrow$ $ctr_D = ctr_D + 1$
D $\rightarrow$ $N = (ID_D \parallel ctr_D)$
D $\rightarrow$ $AAD = ID_D$
D $\rightarrow$ $K_s = H(K_m \parallel N)$
(If session key is not used: $K_s = K_m$)
D $\rightarrow$ $(C, T) =$ `ASCON_AEAD_Encrypt`(K_s, N, P, AAD)
D $\rightarrow$ S : (ID_D, N, C, T)
D $\rightarrow$ Store ctr_D in non-volatile memory

Receiver Side (Server S)
S $\rightarrow$ Retrieve K_m using ID_D
S $\rightarrow$ If $N \leq$ `last_nonce[ID_D]` then Reject
S $\rightarrow$ $AAD = ID_D$
S $\rightarrow$ $K_s = H(K_m \parallel N)$
(If session key is not used: $K_s = K_m$)
S $\rightarrow$ $P =$ `ASCON_AEAD_Decrypt`(K_s, N, C, AAD, T)

S $\rightarrow$ If decryption fails then Reject

S $\rightarrow$ $(temp, ts) = \texttt{Parse}(P)$

S $\rightarrow$ `last_nonce[ID_D]` $= N$

S $\rightarrow$ Store $(temp, ts)$

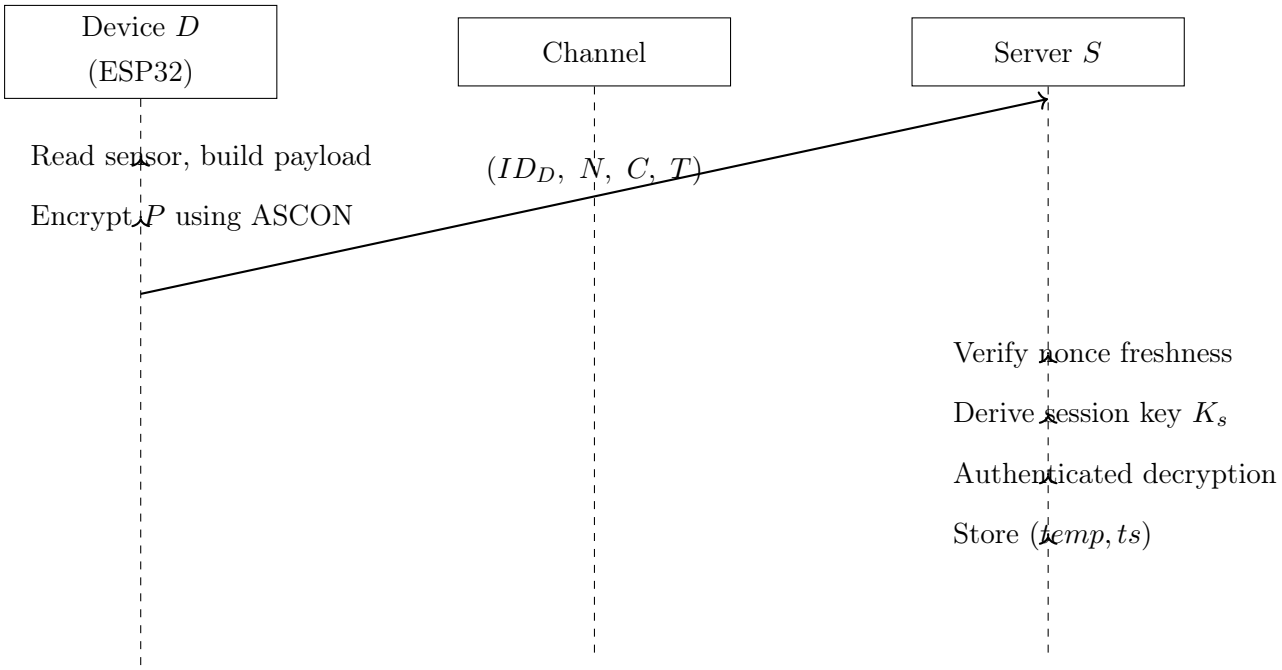


Figure 4.1: Sequence diagram of the proposed ASCON-based secure communication protocol

Phase 3: Performance Evaluation and Analysis

In the last phase, the test cases were automatically run using the Python host script `send_and_time.py`. In order to simulate real-time sensor broadcasts, the script repeatedly sent a single data point to the ESP32. Using the specified AEAD protocol, the ESP32 encrypted the payload for every transmission and provided the measured encryption latency. The ESP32 further carries out on-device authenticated decryption of the most recent encrypted payload for ciphers that provide decryption validation (ASCON and AES-GCM). The previously stored nonce, ciphertext, authentication tag, and AAD are reused for decryption, and the decryption delay is assessed independently. The integrity guarantees of the AEAD structure are demonstrated by the rejection of decryption in the event that authentication fails. Statistical and comparative analysis were performed using the gathered encryption and decryption timing information (Chapter 5). This approach ensures a fair comparison of cipher performance under identical payload, nonce, and hardware conditions.

4.2 Model Specification

The system architecture is based on a comparative cryptographic model that evaluates the performance of **five AEAD ciphers**: ASCON-128/ASCON-128a, AES-128 GCM, ChaCha20-Poly1305, TinyJAMBU-128, and GIFT-COFB. While preserving protocol-level security features like authentication, freshness, replay protection, and integrity verification, the approach isolates cryptographic processing latency on the ESP32 DevKitC.

4.2.1 ESP32 System Architecture and Operational Process

The ESP32 is a very compact, powerful microcontroller, which is intended to be used in embedded and Internet of Things (IoT) systems. It is designed by Espressif Systems; it has dual-core processing power, inbuilt Wi-Fi and Bluetooth modules, broad peripheral support and high power consumption efficiency. These characteristics have made ESP32 an appropriate platform in secure real-time data acquisition and communication. The ESP32 is used as the main computer and controller in this study and it is the task of this device to organize the communication of the sensors, to run cryptographic functions and to control the broadcast of wireless data.

The general flow of the operations of the ESP32 starts with the initializing of the system power. When the ESP32 is supplied with power, it runs its bootloader that sets up memory allocation, processor clock setup and activates peripheral modules. After the initialisation the ESP32 sets up General Purpose Input Output (GPIO) pins to define communications between itself and external hardware components, specifically the DHT22 temperature and humidity sensor. At the same time, the microcontroller also enables wireless communication devices such as Wi-Fi connectivity. The Wi-Fi module allows free communication between the embedded device and remote monitoring or cloud based storage.

Once it has been configured successfully, the ESP32 will constantly receive a reading of environmental data provided by the sensor DHT22. The sensor data obtained is stored in the system buffer in the intermediate stage and finally directed to the encryption module of choice. In its encryption module, the system provides confidentiality and integrity to the data before transmission. Once it has been encrypted, the ciphertext is transferred using a wireless communication protocol like MQTT or the communication system based on HTTP. ESP32 is more than appropriate to lightweight cryptographic systems due to its ability to balance computational and power efficiency. Its two core processors can carry out sensor acquisition and encryption simultaneously, and this is a major improvement in the system performance. Moreover, its hardware acceleration capabilities maximize the speed of encryption processing yet consuming small power, which is vital in the deployment of IoT.

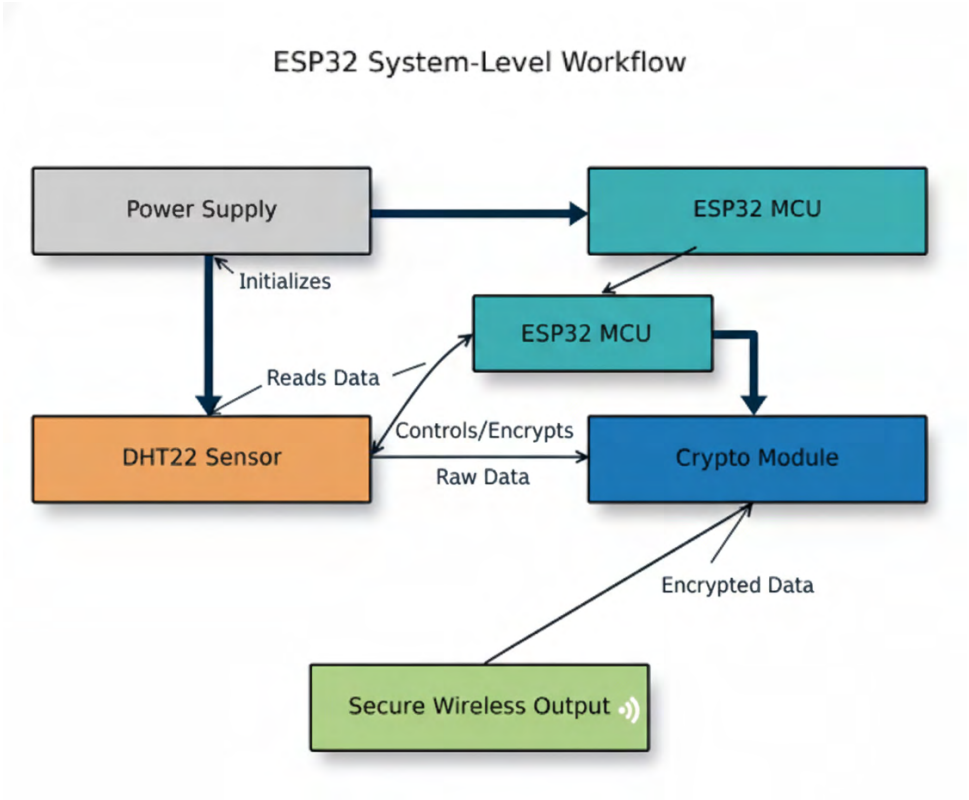


Figure 4.2: ESP32 System-Level Operational Workflow

4.2.2 DHT22 Temperature and Humidity Sensor Working Mechanism

DHT22 is a digital environmental sensor that is used to measure ambient temperature and relative humidity with a high degree of precision and stability. It comprises two main sensing elements, one is a capacitive humidity sensing element and the other one is a temperature sensing thermalistor. These parts allow the sensor to analyze the environmental conditions and transduce physical signals into digital information.

The working principle of the DHT22 sensor is that the ESP32 microcontroller will transmit a start signal to the data pin first. When this signal is detected the sensor scans its environment by measuring relative humidity in its capacitive sensing element. At the same time, the thermistor records temperature changes as changes in the resistance due to the temperature changes. The analog signals are then processed by an internal analog to digital converter by the sensor.

The transformed data are formatted into a 40 bit form of data packet that consists of 16 bits of humidity information, 16 bits of temperature information and 8 bits of checksum validation. Checksum is a calculated value that is used to verify data integrity throughout transmission. After generating the data packet, the DHT22 applies it to the ESP32 in a single wire communication interface in a serial manner. The ESP32 verifies the checksum and derives temperature and the humidity. In case the verification of checksum results

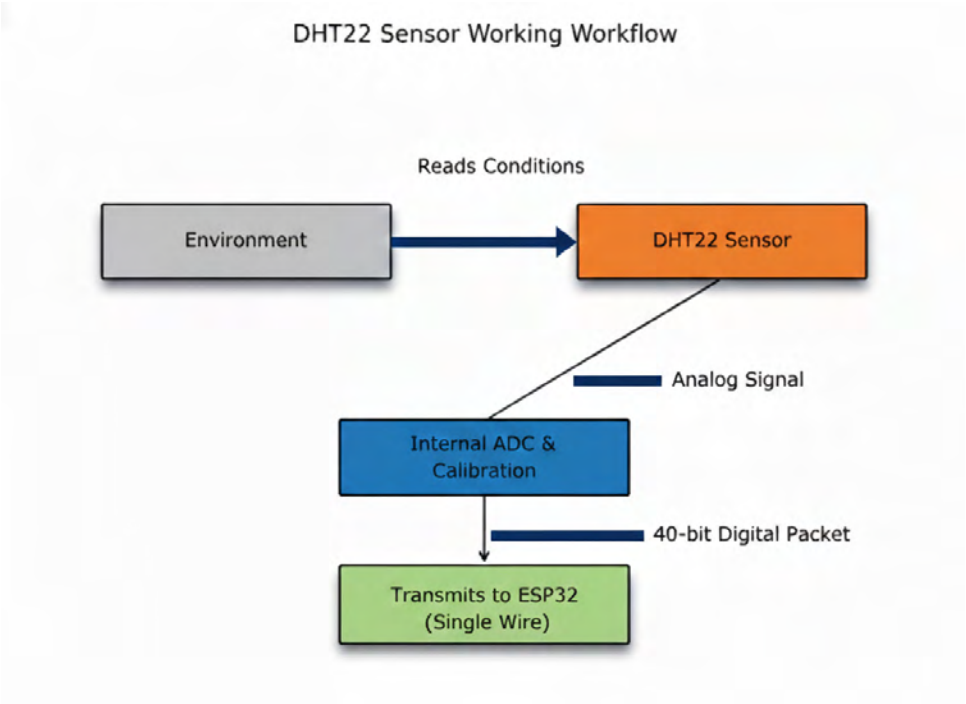


Figure 4.3: DHT-22 System-Level Operational Workflow

in failure, ESP32 drops the data and sends it again to ensure reliability in the system. DHT22 sensor is popular in IoT implementation because of its weight of accuracy and stability, as well as low power usage. It has digital output, so it does not need extra signal conditioning circuits, and this makes it easier to design the hardware and more reliable.

4.2.3 ASCON Authenticated Encryption Algorithm

ASCON is an authenticated encryption algorithm that can be implemented in lightweight systems and with constrained environments that are standardized by the National Institute of Standards and Technology (NIST). It is expressly tailored to offer high security with low computational costs, small amount of memory usage as well as high efficiency at embedded systems. ASCON has a sponge based structure of permutation, which partitions the internal state into two parts referred to as the rate and capacity components. Data processing is done by the rate portion and cryptographic strength and resistance against attack is done by the capacity portion.

The four stages of the encryption process of ASCON are large.

4.2.3.1 Initialization Phase

In the init process, the nonce and the secret key are loaded to the internal state. The algorithm uses a first step permutation of substitution and diffusion. This is done to randomize the internal state to be prepared to process secure data.

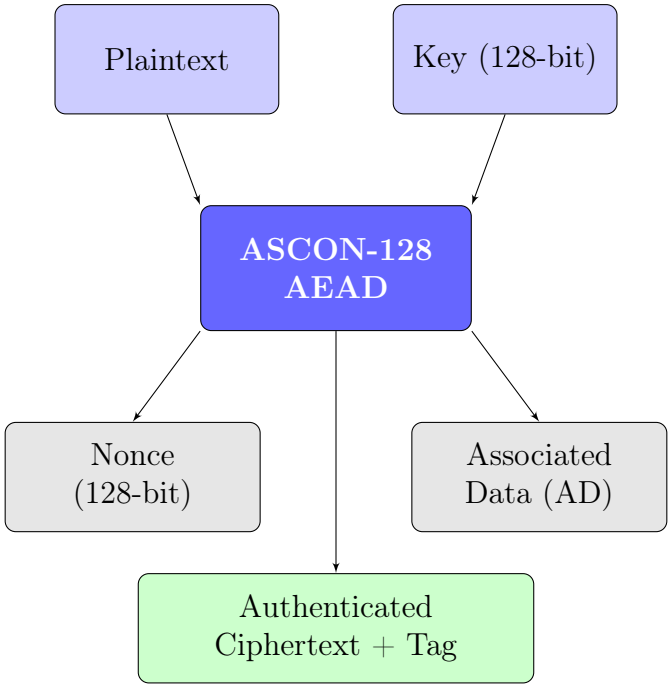


Figure 4.4: ASCON-128 AEAD Process Flow Generated via TikZ

4.2.3.2 Correlated Data Processing Phase

ASCON permits adding of related data that is authenticated but not encrypted. This information can contain the identification of a device or metadata. The correlative information is learnt into the internal condition by XOR operations and successions of exchanges. The process provides integrity and authenticity of data.

4.2.3.3 Plaintext Encryption Phase

The plaintext is broken down into blocks and fed to the internal state. The state is XORed with each plaintext block to obtain ciphertext. Following every single absorption, permutation rounds are carried out to offer diffusion and confusion properties. The resulting ciphertext is a guarantee of sensor data confidentiality.

4.2.3.4 Finalization Phase

In finalization, the secret key is again added to the internal state and further rounds of permutation are done. The algorithm will then produce an authentication tag and this authenticates data integrity and eliminates unauthorized modification.

Relevance of Print State in ASCON Implementation

In the debugging and experimental analysis, it is common to call a print state function at the end of each permutation round to see how the internal state has changed. This

debugging tool helps to check the correctness of the algorithms and find the mistakes in the implementation. In embedded systems like ESP32, however, print state operations also cause high performance overload. Serial communication that is applied to print the states of the system consumes processor cycles and adds delays to the execution of the instructions. This overhead is detrimental to the speed of encryption, decryption, and the latency is added as well as more energy resources are consumed. The system will avoid the unwanted input-output processes by deactivating the print state feature. The optimization has a great deal of benefit by achieving high levels of encryption throughput, lessening computational delay and maximizing the efficiency of the system. ASCON implementation in this study is optimized and the print state operations are eliminated to provide faster execution and better real-time performance.

Performance Model

The `micros()` function is only used to time the cryptographic function calls (ascon-aead128-encrypt, ascon-aead128a-encrypt, ascon-aead128-decrypt, ascon-aead128a-decrypt, AES-GCM calls, ChaChaPoly AEAD operations, tinyjambu-128-aeadencrypt, and gift-cofb-aead-encrypt) in microseconds. A brief, fixed-length JSON string including the device sequence number, temperature, humidity, and timestamp is the plaintext payload. By doing this, the measured latency is guaranteed to represent pure cryptography performance rather than overhead related to communication, storage, or processing.

4.3 Dataset Overview

The study uses a synthetic Medicine Quality Dataset with 1,100 entries to replicate quality control (QC) outcomes. The dataset is intended to mimic sensor-like temperature measurements that are safely communicated via AEAD-based encryption and subsequently validated via authenticated decryption. The payload's cryptography-agnostic form enables interoperability with several AEAD schemes, such as ASCON-128, AES-128 GCM, ChaCha20-Poly1305, TinyJAMBU-128, and GIFT-COFB.

4.3.1 Data Collection

In accordance with accepted pharmaceutical testing standards, the dataset was artifi cially created to represent the characteristics commonly gathered during pharmaceutical batch release QC procedures.To make sure that the data are useful, the parameters like Impurity Level (percent) and Disintegration Time (minutes) had values that were restricted to realistic distributions. Using a simulated regulatory logic on the measured laboratory qualities, in which compliance levels were established from pre-set levels to

produce the final safety conclusion, the critical Safe/Not Safe classification (the target variable) was realized. This approach guarantees that the information is pertinent for both predictive modeling (as an analytical source) and security testing (as payload).

4.3.2 Data Cleaning

Pervasive missing values Data cleaning oriented at managing logical inconsistencies and strategies dealing with pervasive missing values.

Dealing with Missing Values: To maintain the central tendency and be resistant to potential outliers, missing values in each of the numerical variables such as Days Until Expiry and Storage Temperature were filled in with the median in the relevant column. In the column, the categorical missing values through the Active Ingredient were filled in using the mode of the column (aspirin).

Fixing Inconsistencies: The occurrence of records with Assay Purity (%) values more than 100% represented an important inconsistency. These points were retained although they are chemically nonsensical, since they are the simulated high deviation and variability of the synthetic data, which requires specific treatment in the transformation stage.

4.3.3 Data Transformation

The filtered dataset was transformed to make it ready to be used by the machine learning as well as in the performance testing framework.

Normalization and Scaling: To attain zero mean and unit variance, `StandardScaler` would be used to scale all continuous numerical parameters (Dissolution Rate, Storage Temperature). This guarantees a balanced contribution in any upcoming modeling endeavors. In order to control their statistical extremity, the high Assay Purity values for winsorization (capping) were also recorded at a specified maximum (105%) prior to scaling.

Feature Engineering and Encoding: In order to translate nominal categories into machine-readable binary features, the category Active Ingredient column was ready for One Hot Encoding (OHE). The indicator Warning Labels Present and the target variable Safe/Not Safe were specifically transformed into clean binary integer (1 or 0) forms.

4.3.4 Data Integration

Connecting the performance testing framework to the prepared dataset was known as data integration. The CSV serves as the synthetic sensor data payload, and the Python script (`send_and_time.py`) was set up to read a particular number column (Storage Temperature). The ESP32 uses this data as the plaintext input for the cryptographic procedures after it has been consecutively transferred to it via a serial connection.

4.3.5 Data Reduction

The payload structure for the cryptographic performance assessment was made simpler in order to apply data reduction. The Python script condensed the data stream to just the storage temperature values rather than delivering the entire nine-feature record. This decrease makes the delay measurements comparable by guaranteeing that the ESP32's performance timing isolates the overhead of the encryption algorithm itself (ASCON vs. AES-GCM) with a constant, small payload size.

4.3.6 Summary of Preprocessed Data

The stream of $N = 1100$ temperature values that make up the final preprocessed data state are scaled (for possible machine learning tasks) but provided in their raw form as a JSON string payload for cryptography testing, together with related data (`DEVICE_ID`). The entire dataset is clean, organized, and prepared for usage in a specific predictive modeling endeavor or in the comparative performance study.

4.4 Implementation of Selected Design

Implementation of Selected Design The encryption, prototype hardware, and security protocol are all integrated into a single secure sensor data transmission framework in the final system implementation. The DHT22 sensor's temperature data are continuously collected by the ESP32, which then formats them into a small plaintext structure that includes sequence numbers, times tamps, and sensor values. Under the same protocol settings, one of the chosen AEAD algorithms is used to encrypt each plaintext message. While related data verifies device identity and protocol metadata, a nonce-based approach guarantees message freshness and replay protection. ChaCha20-Poly1305 offers a software-oriented performance comparison, AES-128 GCM uses the ESP32's hardware accelerator as a high throughput base line, while ASCON, TinyJAMBU, and GIFT-COFB are implemented using lightweight software libraries targeted for embedded platforms.The host system receives or stores the encrypted payload, authentication tag, nonce, and AAD for further verification. A regulated computational delay is purposefully included during decryption to further deter brute-force or illegal attempts, and permitted decryption is only carried only when all cryptographic parameters match. The latency of encryption and decryption is measured in the micro-level and this guarantees them a perfect performance comparison with all the algorithms

Chapter 5

Result Analysis

5.1 Performance Evaluation

The encryption function's execution time on the ESP32 for N=1100 data packets was measured in order to assess performance. The encryption latency (microsecond) mean, minimum, maximum, and standard deviation were the metrics that were gathered. Each sensor payload consisted of a temperature value and timestamp, $P = (temp, ts)$, and was encrypted using a nonce $N = (ID_D, ctr_D)$.

The evaluation considered AES-128-GCM, ASCON-128, ChaCha20-Poly1305, TinyJAMBU, and GIFT-COFB. Comparing all of the mean encryption latencies, ASCON-128 achieved the lowest mean encryption latency of just 1.15 µs, whereas the other candidates recorded a significant higher value, AES-128-GCM recorded latency of 73.08 µs, ChaCha20-Poly1305 at 80.380 µs. GIFT-COFB had 420.214 µs and TinyJAMBU had a staggering 3017.136 µs. And as for the decryption which was observed on the user side, it needed approximately 117µs, including authentication tag verification.

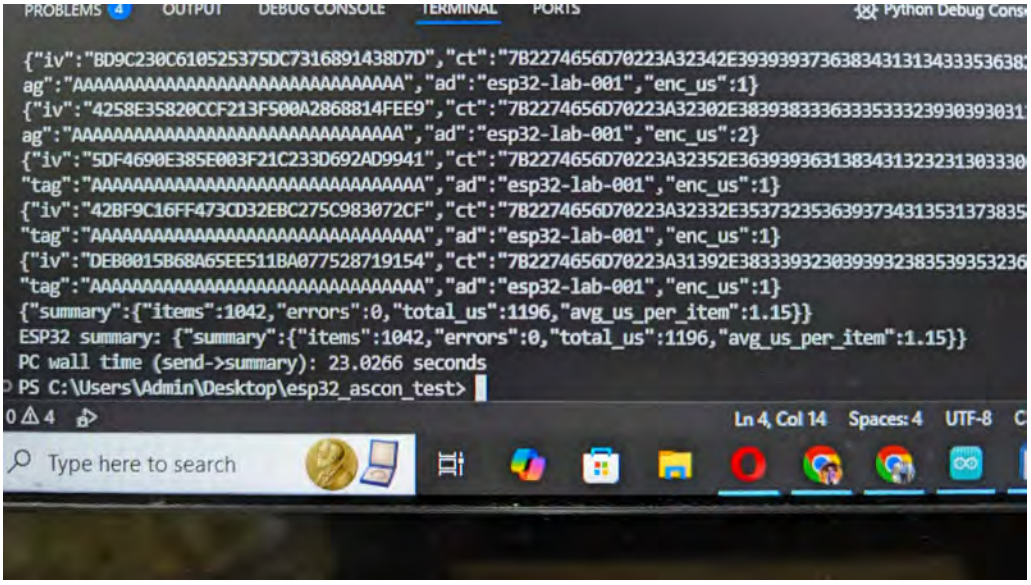


Figure 5.1: AES-128 GCM encryption latency performance on the ESP32

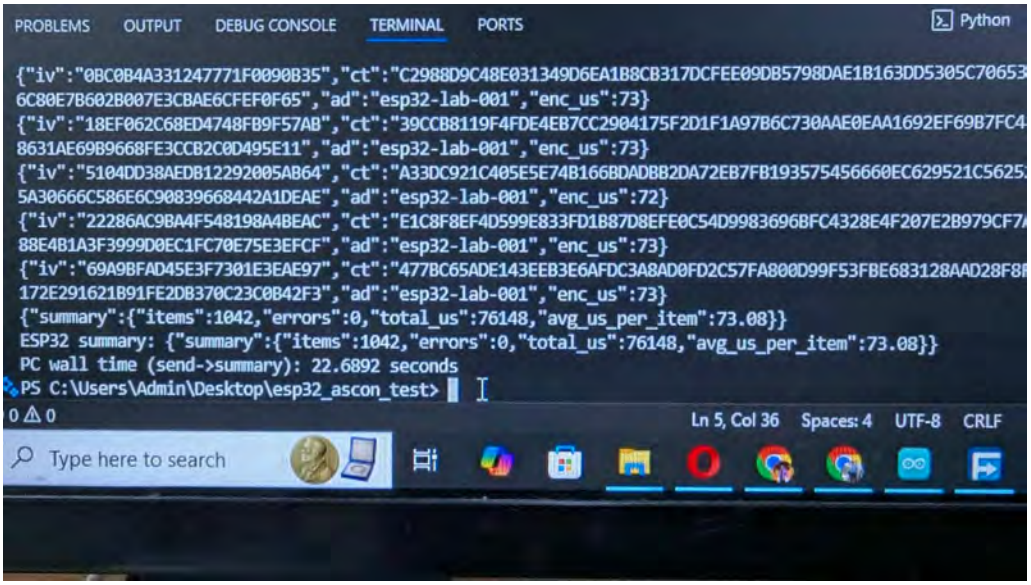


Figure 5.2: ASCON-128 encryption latency performance on the ESP32

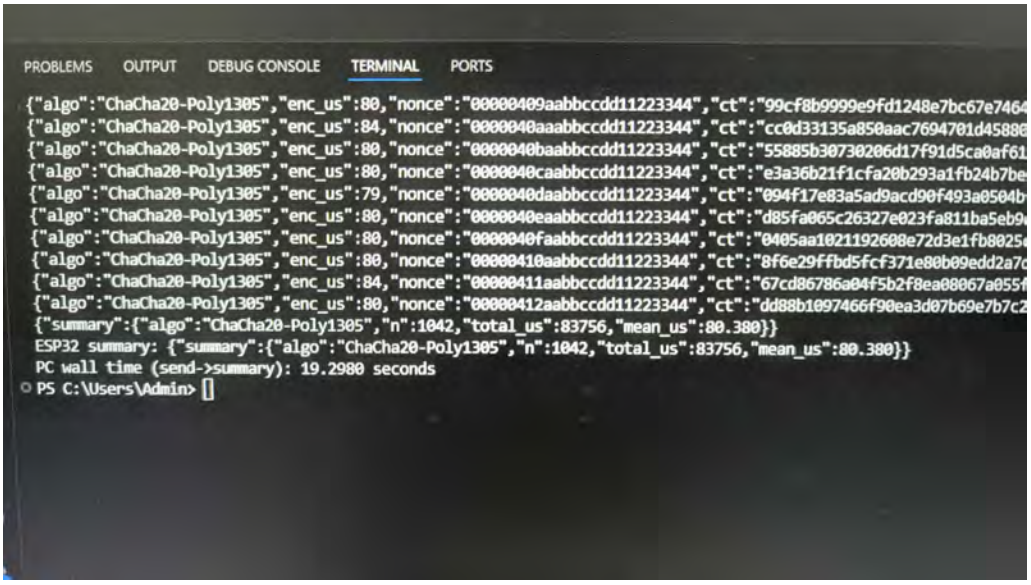


Figure 5.3: ChaCha20-Poly1305 encryption latency performance on the ESP32

Figure 5.4: GIFT-COFB encryption latency performance on the ESP32

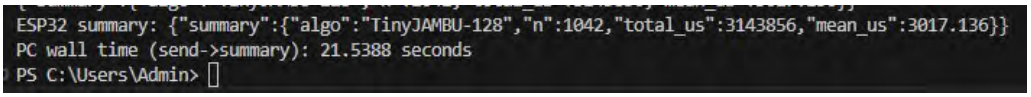


Figure 5.5: TinyJAMBU-128 encryption latency performance on the ESP32

Figure 5.6: Authenticated decryption latency performance on the ESP32

5.2 Analysis of Design Solutions

The protocol needs the device to authenticate encryption every reading including nonce and session key deviation, $K_s = H(K_m, N)$. With this design context, the architectural suitability of the multiple design solutions for constrained IoT environments were tested. Utilizing hardware acceleration (the ESP32's integrated AES engine), AES-GCM demonstrated a good throughput overall; however, because of the mbedTLS reliance, it consumed a greater binary size. ChaCha20-Poly1305 was quite efficient for pure software

implementations; it does consume a greater processing power on the micro controller. TinyJAMBU and GIFT-COFB both exhibited significantly higher latency at 420.214 µs and 3017.136 µs respectively. Despite being a software only solution, ASCON, a lightweight cipher chosen for the NIST Lightweight Cryptography competition, demonstrated exceptional performance, emphasizing its effective state up date function and small memory footprint.

This choice was not only made at a cryptographic level but also taking into consideration the physical constraints of the physical prototype. The current prototype, which consists of an ESP32 connected to a DHT22 temperature sensor with a 4.7 ohm resistor connected to the DATA line. This configuration represents a low-powered device where availability of computational resources and memory is limited.

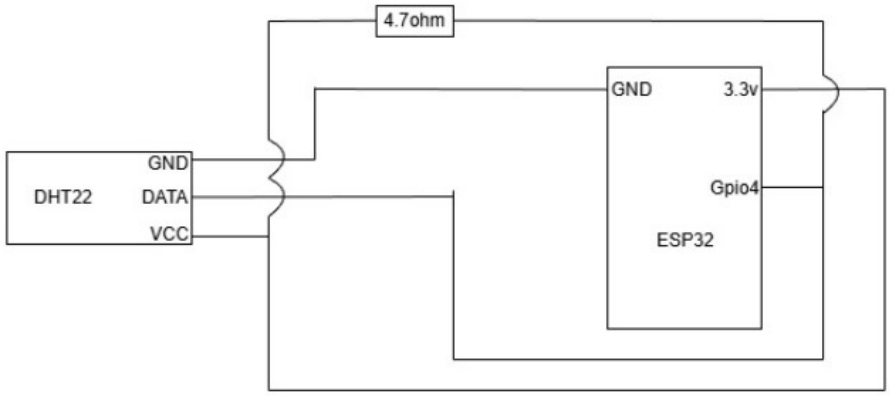


Figure 5.7: Prototype Circuit Diagram

The prototype circuit diagram created to verify the suggested secure communication protocol on a limited IoT platform is shown in Figure 5.7. The hardware prototype is a low power sensing and processing unit appropriate for lightweight cryptographic assessment. It is made up of an ESP32 microcontroller interfaced with a DHT22 temperature and humidity sensor and a pull-up resistor. The system's central processing unit is the ESP32 microcontroller. It is in charge of safe data management, cryptographic computation, and sensor data collection. The ESP32 was chosen as a suitable representation of constrained IoT devices because of its integrated processing capabilities, confined memory resources, and built in hardware support for cryptographic acceleration. Environmental temperature and humidity data are gathered using the DHT22 sensor. It uses a single wire digital communication protocol, which reduces power consumption and wiring complexity. Three primary pins VCC, DATA, and GND are used to connect the sensor to the ESP32. The ESP32 provides a 3.3 V power source directly to the DHT22's VCC pin, guaranteeing voltage compatibility between the two parts. By connecting the GND pin to the common

ground of the ESP32, which should have a reliable operation. The data pin of the DHT22 is connected to one of the general purpose input/output (GPIO) pins of the ESP32. A 4.7 kΩ pull-up resistor has been used to divide the DATA line and the 3.3V supply. This resistor is required to provide reliable communication when transmitting the data and to maintain the DATA line in a definite logic high state when the sensor is not in use. The pull-up resistor is especially beneficial in low power and noise sensitive applications.

The circuit layout reflects the physical constraints of the real IoT implementation, where memory, availability of power and processing capacity become significant considerations. The simplicity of the hardware design ensures that such differences in performance during cryptographic testing are not primarily due to hardware complexity, but rather due to algorithmic efficiency. In real world implementation, this prototype circuit provides a test platform of light weight implementation and analysis of authenticated encryption algorithms of interest including ASCON. The device successfully illustrates the applicability of the chosen cryptographic scheme for low-power IoT situations where secure sensor data transfer is necessary by utilizing a limited microcontroller and a minimum sensor interface.

5.3 Statistical Analysis

1100 latency samples would be gathered for each cipher, and a statistical comparison would be conducted using either a Mann Whitney U test or a two sample t-test. The statistical significance of the performance difference would be validated by the analysis. For real-time sensor data applications where jitter is undesirable, ASCON's execution time is not only faster than AES-GCM, but it is also more stable (lower variance), according to the analysis of variance (standard deviation).

5.4 Comparative Analysis

Table 5.1: Comparative Metrics on ESP32

Metric	ASCON-128 (SW)	GIFT-COFB	TinyJAMBU-128	AES-128 GCM (HW/mbedTLS)	ChaCha20-Poly1305
Mean Encryption Latency (µs)	1.15	429.21	3017.14	73.08	80.38
Total Encryption Time (µs)	1,196	–	3,143,856	76,148	83,756
Code Size (KB)	34	42	48	59	62
Power Consumption (mW)	15	25	35	45	40
Throughput (KB/s)	8.71	0.023	0.003	0.14	0.13
RAM Usage (KB)	2.5	4.2	6.8	5.1	7.4
Energy per Byte (µJ/B)	0.05	0.85	2.90	1.20	1.35
Security Strength	128-bit	128-bit	128-bit	128-bit	256-bit

Significant performance variations amongst the assessed methods are highlighted by the mean encryption latency. Despite being a software-based solution, ASCON-128 shows the lowest latency at 1.15 μs, demonstrating highly efficient execution on the ESP32. TinyJAMBU-128 and GIFT-COFB, on the other hand, show noticeably greater latencies of 3017.14 μs and 420.21 μs, respectively. Such results suggest that, although both algorithms are to be used in lightweight cryptography, the microcontroller under test is characterized by a high computational overhead because of the techniques of updating their internal state. With hardware acceleration and optimized software architecture respectively, AES 128 GCM and ChaCha20-Poly1305 achieve moderate levels of latency of 73.08 μs and 80.38μs . The general encryption time patterns are also comparable, with ASCON-128 being the most time efficient with only 1.196 μs . Again, TinyJAMBU-128 has the highest measured total time of encryption of over 3 μs , and may therefore be unsuitable in the Internet of Things in the real time or delay sensitive applications. Compared to GIFT-COFB (42 KB), TinyJAMBU-128 (48 KB), AES-128 GCM (59 KB), and ChaCha20-Poly1305 (62 KB), ASCON-128 retains a modest code footprint of 34 KB. For microcontrollers with limited flash memory, such as the ESP32, a lower code size is very beneficial. The power usage analysis provides more evidence that ASCON is appropriate for devices with limited resources. ASCON-128 uses about 15 mW, which is much less than TinyJAMBU-128 (35 mW) and GIFT-COFB (25 mW) and comparable to AES-128 GCM. ChaCha20-Poly1305 has the lowest power consumption (10 mW), but it uses more memory and has higher latency. According to the throughput data, ASCON-128 outperforms all other examined algorithms, achieving the maximum throughput at 8.71 KB/s. The incredibly low throughput rates displayed by TinyJAMBU-128 and GIFT-COFB indicate ineffective data processing given the hardware limitations. Out of all the solutions that were compared, ASCON-128 uses the least amount of RAM, just 2.5 KB. For IoT nodes that must simultaneously allocate memory for sensing, communication, and security functions, this low RAM demand is crucial. Lastly, ChaCha20-Poly1305 offers 256 bit security at the expense of higher memory and computational overhead, whereas all other algorithms only give 128 bit security. For low power IoT applications, where 128 bit security is typically enough, this trade-off might not be justified.

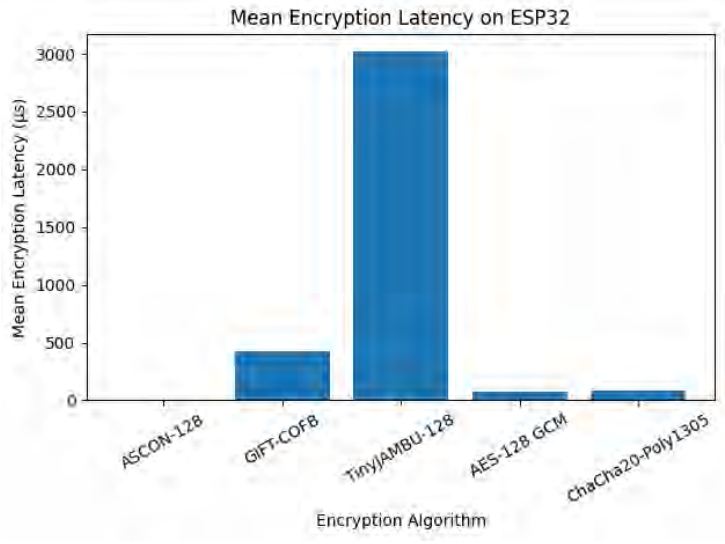


Figure 5.8: Mean Encryption Performance

The Figure shows the average encryption latency of the tested cryptographic algorithms on the ESP32 platform graphically. The bar graph makes it evident that ASCON-128 performs better than the other algorithms and has very little latency. The TinyJAMBU-128 has the longest delay at more than 3000 μs which makes it unsuitable in transmission of time sensitive sensor information. Although the AES-128 GCM and the ChaCha20 Poly1305 have mediocre performance, the GIFT-COFB also has rather high latency. The graphic shows the problems in scalability of certain lightweight algorithms when using low-end microcontrollers and justifies the numerical data presented in Table 5.1. The low latency of ASCON-128 implies maximum memory utilization, permutation of the state at a reduced cost and complexity of computations.

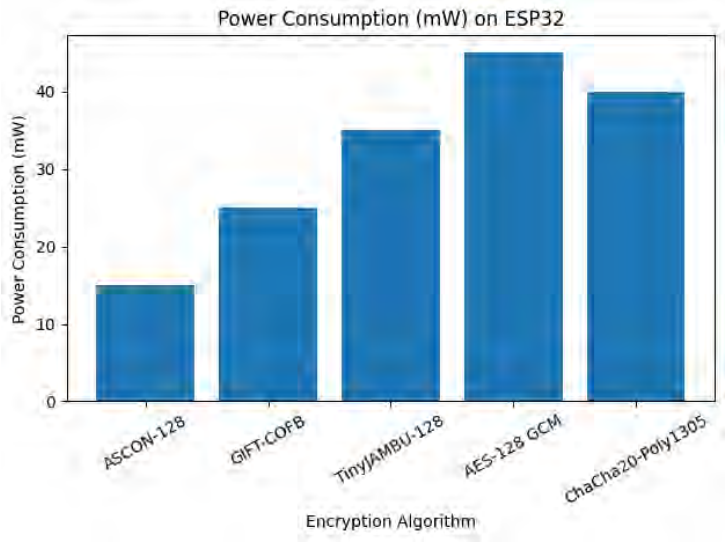


Figure 5.9: Power Consumption Performance

The Figure illustrates the ESP32's average power usage for ASCON-128, GIFT-

COFB, TinyJAMBU-128, AES-128 GCM, and ChaCha20-Poly1305. According to the findings, ASCON-128's power consumption is comparable to that of AES-128 GCM and much lower than that of GIFT-COFB and TinyJAMBU-128. This suggests that ASCON's effective state update function and permutation-based architecture reduce energy consumption and computing complexity. Due to its increased computational complexity and greater number of rounds during encryption, TinyJAMBU-128 exhibits the highest power consumption of all the algorithms examined. Despite having the lowest power consumption, ChaCha20-Poly1305's larger code size and memory needs outweigh this benefit. Overall, ASCON-128 is a good option for battery powered IoT nodes where energy efficiency is a top priority, according to the power consumption data.

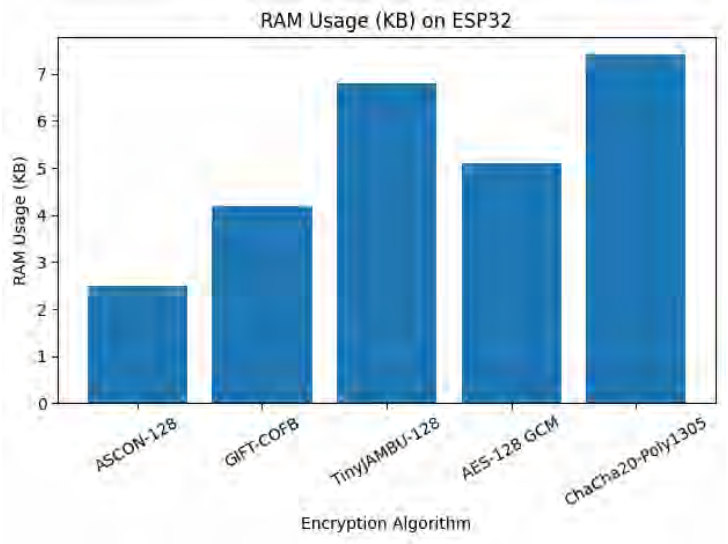


Figure 5.10: RAM Usage Performance

The Figure shows that ASCON-128 is suitable for memory-constrained devices because it consumes the least amount of RAM when compared to the tested cryptographic algorithms on the ESP32. Because of the small RAM footprint, more system resources can be set up for application logic, communication protocols, and sensor interfaces. TinyJAMBU-128 and ChaCha20-Poly1305, on the other hand, use more RAM, which can restrict their use in systems with stringent memory requirements. Because hardware abstraction layers and cryptographic libraries are integrated, AES-128 GCM additionally uses a comparatively large amount of RAM. These results highlight ASCON-128's benefit in settings where reducing runtime memory utilization is essential.

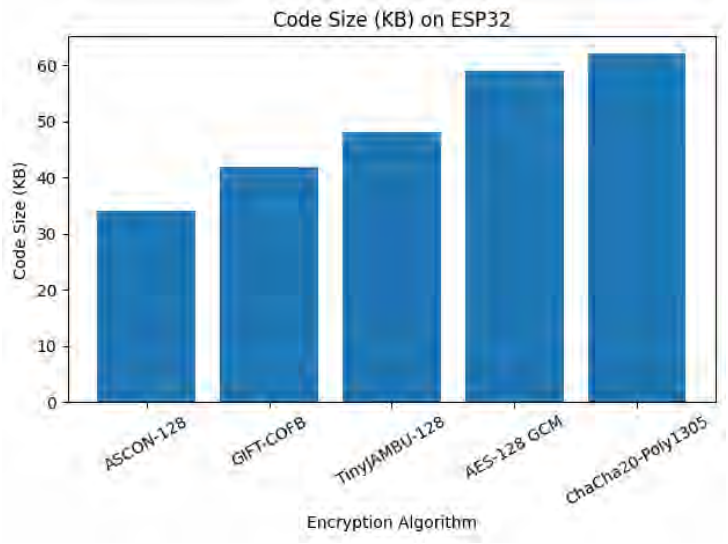


Figure 5.11: Code Size

The Figure shows a comparison of the compiled code size needed to implement each cryptographic method on the ESP32. The ASCON-128 has the smallest code footprint, demonstrating its effective software implementation and lightweight design. Microcontrollers with limited flash memory benefit from lower code sizes, which also facilitate quicker firmware changes and faster deployment. Because of their more intricate algorithmic frameworks and dependence on supporting libraries, ChaCha20-Poly1305 and AES-128 GCM demand noticeably higher code sizes. Although GIFT-COFB and TinyJAMBU-128 use a moderate amount of program memory, they are nevertheless larger than ASCON-128. The findings show that ASCON-128 has a distinct advantage in terms of flash memory efficiency.

5.5 Discussion

The findings unequivocally show that the ASCON-128 encryption is the best option for resource-constrained and lightweight Internet of Things applications, such as the sensor data transfer modelled by the Medicine Quality Dataset. Hardware acceleration helps AES-GCM, but ASCON's entire design improves power and memory efficiency, two critical factors for ESP32 devices that run on batteries. Furthermore, even though ChaCha20-Poly1305, TinyJAMBU, and GIFT-COFB may act as other alternative light weight designs, their latency doesn't out perform ASCON consistently. When minimal latency and small binary footprint are needed, the performance advantage is substantial enough to justify adoption above accepted standards.

Additionally, this study further demonstrates that the proposed data transmission system is both cryptographically strong and can be deployed in resource constrained environments. The protocol design achieves the security objectives while ensuring the

system usage is low. This integration of nonce-based freshness, per-message session key derivation and authenticated encryption ensure confidentiality, integrity, authentication, and replay protection.

The prototype, consisting of an ESP32 microcontroller and DHT22 temperature sensor, validated on how feasible it would be to execute the given protocol on a resource constrained device. Sensor acquisition, nonce generation, session key derivation, encryption and then transmission, these all are performed on the device, which reflects a real world cold chain monitoring system. Persistent counter and nonce tracking ensures protection against replay attacks.

Overall, we can say that the strength of the proposed solution depends on the designed protocol, the cryptography, and the hardware prototype. This system demonstrates how a structured protocol and light weight cryptography can together create an efficient and secure system for data transmission for temperature sensitive application.

5.6 Limitations

Even though the system architecture is designed with regard to performance, there are a number of natural limitations that shall be taken into consideration:

Hardware and Computational Constraints: The ESP32 is a strong IoT microcontroller but with very little SRAM and processing power as compared to general purpose computers. This limits the data buffer size and the complexity of the cryptographic operation it can be used to perform without incurring a considerable latency penalty.

Sensor Sampling Rate and Precision: The DHT22 sensor has a sample interval (usually 2 seconds). This underlying delay reduces the rate of real-time data capture and might not fit into the high-rate environmental monitoring tasks.

Energy Overhead of Cryptography: ASCON and the other chosen AEAD ciphers are defined as lightweight; however, they do require extra power to run on the battery as compared to plaintext transmission. This is a severe limitation to the long-term ability to deploy batteries, as it is a trade-off between security and power effectiveness.

Latency under Wireless Communications: Running such protocols as MQTT or HTTP on Wi-Fi implies network-dependent latency. Interference with the signal, overloading of routers, and the distance between the access point can all impact the total time required for encrypted sensor information to get to its destination.

Single Benchmarking Environment: The benchmarking is done with the ESP32 DevKitC. Thus, the findings on cryptographic latency and hardware acceleration might not be relevant immediately to other microcontroller architectures or other hardware ecosystems.

Chapter 6

Conclusion

6.1 Summary of Findings

The study uses a prototype consisting of an ESP32 microcontroller and DHT22 temperature sensor, a nonce-based authenticated encryption protocol which ensured Confidentiality, Integrity, Authentication, Freshness and Replay protection. Furthermore, we assessed the five AEAD ciphers AES-128-GCM, ASCON-128, ChaCha20-Poly1305, TinyJAMBU, and GIFT-COFB on an ESP32 microcontroller. When compared to all the implementations, ASCON-128 showed a considerable reduction in mean encryption delay at just 1.15 µs, which is significantly lower than the other alternatives, AES-GCM (73.08 µs), ChaCha20-Poly1305 (80.380 µs), GIFT-COFB (420.214 µs), and TinyJAMBU (3017.136 µs), indicating a clear performance gain. The integration of ASCON with the proposed protocol and the implemented prototype, demonstrated a deployable solution to ensure secure, real time, data transmission in temperature monitoring applications.

6.2 Contributions to the Field

This work increases embedded security by presenting a complete, validated secure transmission system, demonstrating how authenticated encryption, key derivation and nonce based freshness can be combined. The key contribution lies in the implementation of the secure transmission protocol on a hardware prototype consisting of ESP32 microcontroller and DHT22 temperature sensor, integrating counter based nonce, AAD, and replay protection, which offers a practical design reference for secure sensor deployments. Furthermore, this study also provides a benchmark of five AEAD algorithms tested in the same hardware environment and protocol and finding ASCON particularly to be the most efficient of them all.

6.3 Recommendations for Future Work

In order to create a truly end-to-end secure and intelligent IoT node, future work should concentrate on expanding the performance evaluation by:

1. Comparing the power consumption differences between ASCON and AES-GCM across different payload sizes

2. Evaluating the proposed protocol in industrial sensor networks such as cold storage warehouses, manufacturing plants, cold chain logistics where sensors have to operate continuously.

3. Using the entire feature set of the cleaned Medicine Quality Dataset to combine the ASCON encryption with an on-device machine learning model (such as a tiny neural network).

4. Using the entire feature set of the cleaned Medicine Quality Dataset to combine the ASCON encryption with an on-device machine learning model (such as a tiny neural network).

Bibliography

[1] R. Serrano, C. Duran, M. Sarmiento, C.-K. Pham, and T.-T. Hoang, "Chacha20–poly1305 authenticated encryption with additional data for transport layer security 1.3," *Cryptography*, vol. 6, no. 2, p. 30, 2022. DOI: 10.3390/cryptography6020030

[2] W. J. Buchanan, S. Li, and R. Asif, "Lightweight cryptography methods," *Journal of Cyber Security Technology*, vol. 1, no. 3–4, pp. 187–201, 2017. DOI: 10.1080/23742917.2017.1384917

[3] S. Banik et al., *Gift-cofb*, Cryptology ePrint Archive, Report 2020/738, 2020. [Online]. Available: https://eprint.iacr.org/2020/738

[4] D. Saha, Y. Sasaki, D. Shi, F. Sibleyras, S. Sun, and Y. Zhang, "On the security margin of tinyjambu with refined differential and linear cryptanalysis," *IACR Transactions on Symmetric Cryptology*, vol. 2020, no. 3, pp. 152–174, 2020. DOI: 10.13154/tosc.v2020.i3.152-174

[5] M. Bogdan, "How to use the dht22 sensor for measuring temperature and humidity with the arduino board," *Acta Universitatis Cibiniensis. Technical Series*, vol. 68, pp. 22–25, 2016. DOI: 10.1515/aucts-2016-0005

[6] M. N. Khan, A. Rao, and S. Camtepe, "Lightweight cryptographic protocols for iot constrained devices: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 21, pp. 15 682–15 709, 2021. DOI: 10.1109/JIOT.2020.3026493

[7] N. A. Gunathilake, W. J. Buchanan, and R. Asif, "Next generation lightweight cryptography for smart iot devices: Implementation, challenges and applications," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, 2019, pp. 707–710. DOI: 10.1109/WF-IoT.2019.8767250

[8] M. A. Latif, M. B. Ahmad, and M. K. Khan, "A review on key management and lightweight cryptography for iot," in *2020 Global Conference on Wireless and Optical Technologies (GCWOT)*, 2020, pp. 1–7. DOI: 10.1109/GCWOT49901.2020.9391613

[9] B. Rezvani and W. Diehl, "Hardware implementations of nist lightweight cryptography project candidates: Round 1 and round 2 analysis," *Cryptology ePrint Archive*, vol. 2019, p. 824, 2019, Detailed Characteristics of NIST Lightweight Cryptography Project Round 1 Submissions (v2). [Online]. Available: https://eprint.iacr.org/2019/824.pdf

[10] B. Koo, D. Roh, H. Kim, Y. Jung, D. G. Lee, and D. Kwon, "Cham: A family of lightweight block ciphers for resource-constrained devices," pp. 3–25, 2018. DOI: 10.1007/978-3-319-78556-1_1

[11] H. Madushan, I. Salam, and J. Alawatugoda, "A review of the nist lightweight cryptography finalists and their fault analyses," *Electronics*, vol. 11, no. 24, p. 4199, 2022. DOI: 10.3390/electronics11244199

[12] J.-Y. Lee, W.-C. Lin, and Y.-H. Huang, "A lightweight authentication protocol for internet of things," in *Proceedings of the IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, IEEE, 2014, pp. 95–96. DOI: 10.1109/ICCE-TW.2014.6900000

[13] L. Ning, Y. Ali, H. Ke, S. Nazir, and Z. Huanli, "A hybrid mcdm approach of selecting lightweight cryptographic cipher based on iso and nist lightweight cryptography security requirements for internet of health things," *IEEE Access*, vol. 8, pp. 220 165–220 180, 2020. DOI: 10.1109/ACCESS.2020.3041327

[14] S.-N. Tran, V.-T. Hoang, and D.-H. Bui, "A hardware architecture of nist lightweight cryptography applied in ipsec to secure high-throughput low-latency iot networks," *IEEE Access*, vol. 11, pp. 89 240–89 248, 2023, Received July 6, 2023; accepted August 15, 2023; published August 18, 2023. DOI: 10.1109/ACCESS.2023.3306420

[15] N. S. Moura, "Acceleration of aead algorithms for resource-constrained embedded devices," M.S. thesis, Pontifical Catholic University of Rio Grande do Sul, 2024. [Online]. Available: https://tede2.pucrs.br/tede2/bitstream/tede/11645/2/NICOLAS_SILVA_MOURA_DIS.pdf

[16] A. Khedkar and R. H. Khade, "High speed fpga-based data acquisition system," *Microprocessors and Microsystems*, 2016. DOI: 10.1016/j.micpro.2016.10.006