National College of Ireland

A Hybrid Lightweight Cryptographic Framework with Parallel Processing on Cloud for Secure and Low-Latency IoT Data Transmission

MSc Research Project
MSc Cloud Computing

Sai Gowtham Emani
Student ID:23337818

School of Computing
National College of Ireland

Supervisor: Shreyas Setlur Arun

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Sai Gowtham Emani
Student ID:	23337818
Programme:	Msc Cloud Computing **Year:** 2025-2026
Module:	Msc Research Project
Supervisor:	Shreyas Setlur Arun
Submission Due Date:	02-01-2026
Project Title:	A Hybrid Lightweight Cryptographic Framework with Parallel Processing on Cloud for Secure and Low-Latency IoT Data Transmission
Word Count:	7226 **Page Count** 22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: .....

Date: 02-01-2026

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Hybrid Lightweight Cryptographic Framework with Parallel Processing on Cloud for Secure and Low-Latency IoT Data Transmission

Sai Gowtham Emani

23337818

Abstract

Modern cloud platforms have become an important backbone for processing and securing large volumes of IoT-generated data which applies traditional cryptographic techniques to increase latency and inefficient resource utilisation. To solve this problem, IoT systems have increasingly supported latency-sensitive and security applications. This study has proposed a hybrid lightweight cryptographic framework that uses ASCON at the IoT layer with ChaCha20 at the cloud layer which is increased through parallel cloud processing. The framework has been designed to balance strong security guarantees with low latency and high throughput by reducing limitations of traditional cloud encryption approaches. Python-based IoT simulator and AWS cloud infrastructure has been used to implement and evaluate the system under controlled experimental conditions. There are some results which show that the proposed ASCON–ChaCha20 hybrid framework achieves reduced encryption and decryption time, improved throughput, lower CPU utilization and higher diffusion quality as compared to ASCON–DES and traditional methods. The high avalanche effect confirms enhanced resistance to cryptanalytic attacks.

Keywords: Hybrid Cryptography, Lightweight Encryption, IoT–Cloud Security, Parallel Processing, Performance Evaluation

1 Introduction

The chapter introduces the background and motivation of the study as it discusses the issue of security, latency and performance in the IoT-cloud environment and limitations of traditional cryptographic means. It also states the research question, and objectives, which are a hybrid lightweight cryptography architecture with parallel processing on cloud to enhance the data security and the performance of the IoT.

1.1 Background

The fast development of Internet of Things (IoT) devices has resulted in a greater volume of data creation and cloud-based data processing (Mishra and Tyagi, 2022) that have never been

experienced before. Such devices can be frequently used in resource limited settings that have small computational power, memory, and energy capabilities and the implementation of traditional cryptographic algorithms can be difficult (Ramakrishna and Ali Shaik, 2025). Although the cloud platform provides scalable storage and computation, the transfer of sensitive IoT information to the cloud brings about serious security, privacy, and latency risks (Abba Ari et al., 2024). Although safe, conventional encryption algorithms which may also give computational costs and latency to real-time internet of things applications and, as a result, to system responsiveness and end-user experience.

This study has been motivated by the necessity to achieve a balance between a high level of security and low-latency and high-efficiency in IoT-cloud ecosystems. Light weight cryptographic algorithms like ASCON and ChaCha20 have also become potential alternatives as a result of their lower complexity of computation and better performance on resource constrained gadgets (Patel and Cherukuri, 2025). Nevertheless, using one encryption method might not deliver the best diffusion and resistance against various threat situations. This study is driven to investigate a hybrid lightweight cryptography framework that would be used to complement parallel cloud processing to improve diffusion quality, throughput as well as reduce latency. The study looks to add practical and scalable security solution to the current IoT applications by systematically analyzing the metrics of performance in a cloud-based environment.

1.2 Importance of the Study

The importance of this study mainly lies in solving the growing security and performance challenges of IoT–cloud environments where resource-constrained devices require strong protection without having high latency. There are some traditional cryptographic schemes which does have computational overhead by making them unsuitable for real-time IoT applications (Justindhas and Jeyanthi, 2023). This study has proposed a hybrid lightweight cryptographic framework which has combined with parallel processing to increase and improve data confidentiality, diffusion quality and system performance. By systematically evaluating latency, throughput, CPU utilization and avalanche effect the study provides practical insights into achieving a balanced trade-off between security strength and performance which contributes to scalable and secure IoT data transmission.

1.3 Research Question

What is the role for a hybrid lightweight cryptographic framework using ASCON with ChaCha20 and parallel processing on cloud to improve performance for IoT data security as compared to traditional AES- and DES-based cloud encryption systems?”

1.4 Research Objectives

The research objectives of this study are as follows:

1. To establish a hybrid lightweight cryptographic framework using ASCON with ChaCha20 for secure data transmission in IoT–cloud environments.
2. To quantitatively characterize the behaviour and performance of the proposed framework in terms of evaluation metrics and compare it with traditional AES- and DES-based approaches.
3. To investigate the performance of parallel cloud-based encryption processing in improving real-time performance and scalability for resource-constrained IoT systems.

1.5 Structure of the Report

This report has systematically structured to clearly present the research work and its outcomes. Chapter 1 shows the introduction of the study which includes the background, motivation, objectives and research question. Chapter 2 provides a detailed review of existing literature which is related to IoT–cloud security, lightweight cryptography and hybrid cryptographic frameworks while identifying research gaps. Chapter 3 explains the adopted research methodology, simulation environment, algorithm selection and evaluation parameters. Chapter 4 describes the design specifications and overall system architecture of the proposed framework. Chapter 5 discusses the implementation details, tools used, cloud deployment, and parallel processing mechanisms. Chapter 6 focuses on performance evaluation and experimental analysis. At last Chapter 7 summarizes the findings and presents future work direction

2 Related Work

The chapter is a review of the available cloud-based, lightweight and hybrid cryptography solutions to IoT cloud environments by showing their security, performance attributes and constraints. It also points out the research gaps of concern to the latency, diffusion optimization, system complexity and scalability that drive the proposed hybrid lightweight cryptographic framework.

2.1 Cloud-Based Cryptographic Systems for IoT

The use of cloud-based cryptographic systems is common in an IoT setup because it gives centralized security control, scalable computational capability and secure data storage (Qasem et al., 2024). The main role of the IoT devices in this type of architecture is that of being a data generator by sending sensor data to cloud platforms where it undergoes encryption and decryption operations which is along with management of keys (Javadpour et al., 2024). This can be easily implemented on the device-side and enables the application of powerful cryptographic functions without the audience being restricted by the limited resources of IoT nodes. The cloud providers have automatic scaling which allows cryptographic services to process high amounts of data created by distributed IoT devices.

Nonetheless, using cloud-based encryption will cause issues with latency, bandwidth reliance and real-time responsiveness. IoT transmission of data to the cloud and vice versa extend the end-to-end processing time which may impact time-intensive applications. Also, persistent

connectivity needs render cloud-based cryptographic systems ineffective in distant or unreliable network infrastructures. Although cloud infrastructure may be very powerful in terms of computational power and centralized control, the performance constraints define the necessity of device-level security (Almutairi and Sheldon, 2025). Consequently, the use of lightweight cryptographic methods in conjunction with cloud-based ones is becoming more common to provide scalable, efficient, and secure protection of IoT data.

2.2 Security–Performance Trade-off in IoT Cryptographic Designs

The IoT cryptographic design is associated with a trade-off between high security and the performance of the system. Strong cryptographic algorithms can demand more processing power, more memory and cannot be afforded by the resource-limited IoT devices (Khan et al., 2021). Lightweight encryption methods on the other hand put more focus on speed and efficiency but can have lesser diffusion strength or less protection against advanced cryptographic attacks.

The performance requirements of low latency, low power and high data transmission rate are important in practical IoT application so that real-time application can be supported by low power and high-performance requirements. A large amount of cryptography can result in slower response times, higher power consumption and shorter lifetime of the device (Potlapally et al., 2006). On the other hand, security systems that are simplified to extreme levels can leave sensitive data vulnerable to attacks when being transmitted and saved. To solve this problem, hybrid cryptographic methods spread security burdens across the various layers of the system. The lightweight encryption guarantees the high speed processing of the IoT layers, and the more robust cryptographic functions are done by cloud resources with greater abilities (Dhanda et al., 2020). Such a stratified approach allows a compromise to be achieved where the security levels remain acceptably low but operations are not affected.

2.3 Hybrid Cryptographic Approaches in IoT–Cloud Environments

Hybrid cryptographic methods have been developed as a sound answer to the various security challenges experienced by the IoT -cloud environment such as confidentiality, integrity, authentication, and effective key management. (Pothireddy et al., 2024) suggested a proposal of a hybrid model based on Fully Homomorphic Encryption (FHE) and SHA-3 to encrypt sensitive data on a cloud and ensure its integrity by processing the encrypted data as shown in Table 1. This method has a high efficiency in encryption and decryption than the traditional methods but has several challenges including high computational workload and incompatibility with hardware.

(Farshadinia et al., 2025) has been introduced an IoT-cloud architecture that is energy efficient, incorporating ECDSA, the Short Signature (ZSS) and SEPAR, and introduces new modules, including a centralized third-party auditor and lightweight hybrid encryptor. This model lowers the cost in terms of power use and time but still guarantees mutual authentication, traceability, and multilayer security but raises the complexity of the system and reliance on third-party auditing.

(Ahmad et al., 2023) is proposed to be used with AES (encryption/decryption) to establish a new approach to key management, which is HCA-KMS, in cloud healthcare systems. The model enhances security, lessens time complexity and overcomes limitations of AES key exchange but the matter of scalability and high-complexity of hybridization has been raised. (Farooq et al., 2023) suggested a four-stage IoT-cloud security architecture which incorporates mutual authentication, key distribution, and hybrid encryption, decreasing the encryption and decryption times and throughput, with minimal use of resources, still, it can be problematic in heterogeneous deployments and reliance on fog infrastructure.

Study given by (Ranganatha Rao and Sujatha, 2023) presented Hybrid Elliptic Curve Cryptography (HECC) as the public cloud security scheme, which relies on Edwards curves, IBE, key reduction and AES as the speedy key generation, encryption, and decryption techniques, but the analysis of resisting advanced attacks and large-scale implementation is not the strongest aspect. The study given by (Emre Erbil et al., 2025) deals with embedded systems based on X25519 to perform key exchange, BLAKE3 to perform key derivation, and Ascon-128a to perform AEAD based-encryption to guarantee forward secrecy, integrity, and confidentiality in the resource-constrained device; nevertheless, further experimental and optimization should be done in real-world application.

Lastly, (Muhammed et al., 2025) uses ChaCha20 in addition to ECDH in a hybrid cloud architecture to achieve the best speed, security and scalability, but its use in resource-constrained environments and mass deployment remains to be further confirmed. Taken together, these papers show that hybrid cryptographic solutions are able to manage security, performance, and scalability across IoT of-cloud systems, and such issues as computational overhead, system complexity, hardware dependency, and scalability remain.

Table 1: Summary Table

Study	Hybrid Technique	Key Components	Results / Performance	Limitations
(Pothireddy et al., 2024)	FHE + SHA-3	Fully Homomorphic Encryption, SHA-3	Improved encryption/decryption, secure processing of encrypted data	High computational overhead, hardware limitations
(Farshadinia et al., 2025)	ECDSA + ZSS + SEPAR	Central TPA, HF-Audit, lightweight hybrid encryptor	Reduced power by 30%, time cost by 40%, latency 56 ms, throughput 2144-bit	Increased complexity, reliance on third-party auditing
(Ahmad et al., 2023)	ECC + AES	ECC for key generation, AES for encryption/decryption	Faster encryption, improved confidentiality, integrity, O(n)	Hybrid complexity, scalability challenges

			encryption, O(log n) decryption	
(Farooq et al., 2023)	Hybrid encryption + mutual authentication + key distribution	Novel key distribution, mutual authentication, hybrid encryption	Encryption time reduced by 28%, decryption by 32%, improved throughput	Fog infrastructure dependency, heterogeneous deployment challenges
(Ranganatha Rao and Sujatha, 2023)	HECC + AES + IBE + Diffie-Hellman	Edwards curve key generation, IBE, key reduction, AES	Key gen: 0.000025 s, Encryption: 0.00349 s, throughput: 693.10 kB/s	Limited attack evaluation, large-scale deployment not tested
(Emre Erbil et al., 2025)	X25519 + BLAKE3 + Ascon-128a	X25519 key exchange, BLAKE3 derivation, AEAD encryption	High security, integrity, forward secrecy, lightweight	Limited real-world testing, scaling optimization needed
(Muhammed et al., 2025)	ChaCha20 + ECDH	ChaCha20 encryption, ECDH key exchange	Encryption time 2 ms, key generation 15.8 ms, moderate memory usage	Performance in IoT/mobile not fully validated, large-scale deployment

2.4 Research Gaps Identified

Previous studies on hybrid IoT–cloud cryptographic systems mainly focus on improving security or reducing computation overhead but overlook end-to-end latency and diffusion optimization in real-time cloud-assisted environments. There are quite a few solutions which introduced high system complexity, hardware dependency or trust on third-party components by limiting scalability and practical deployment. Also limited attention has given to parallel encryption execution and comparative evaluation of lightweight cipher combinations under cloud workloads. This study solves these gaps by proposing a parallel hybrid lightweight cryptographic framework that optimizes latency, throughput and diffusion quality while maintaining scalability and deployability in resource-constrained IoT–cloud systems.

3 Methodology

The chapter describes the experimental research design and quantitative design that was used to carry out and test a hybrid lightweight cryptographic framework within an IoT-cloud setup. It describes the simulation environment, choice of cryptographic algorithm, research process

and performance measurement criteria employed to determine the latency, throughput, CPU usage, diffusion quality and encryption/decryption efficiency.

3.1 Research Design and Overall Approach

This study adopts a controlled experimental methodology supported by quantitative analysis to examine the performance of a hybrid cryptographic framework within IoT–cloud environments. The study strategically distributes cryptographic operations by applying lightweight encryption mechanisms at the IoT layer. A controlled simulation-based environment has used. Such performance and security properties as latency, throughput, diffusion, and CPU utilization are quantified at different workloads (Biswas et al., 2024). Hybrid and standalone cryptographic implementations are compared and contrasted to ascertain the efficacy of the given approach in enhancing efficiency and diffusion amount in cloud-based IoTs.

3.2 IoT Simulation Environment

Python was used to make a client-side simulation environment of the IoT (Barriga et al., 2022) that mimics the behavior of the resource-constrained devices in the environment. The simulator creates artificial sensor measurements and uses the lightweight cryptographic functions like ASCON and ChaCha20 to simulate the real life encryption of the IoT. This method does not require any physical hardware and allows to be experimented with controlled conditions of different data sizes and workloads. The simulator captures the encryption time, CPU usage and output features and then sends the encrypted data to the cloud layer.

3.3 Cryptographic Algorithm Selection and Justification

The cryptographic algorithms have been selected for this study which have chosen based on their security strength, computational performance and suitability for IoT and cloud environments. Since IoT devices have operated under some strict resource constraints, lightweight algorithms which have prioritized at the client side while stronger and computationally intensive algorithms have been deployed at the cloud layer.

3.4.2 DES (Baseline Comparator)

This study has included Data encryption standard (DES) as a baseline comparator. Despite being considered insecure by modern standards DES gives simplicity and low computational overhead by making it useful for performance comparison purposes. Its presence enables one to test the performance of the old cryptographic algorithms against the current lightweight and secure cryptographic algorithms. Compared to DES, this study outlines how more recent cryptographic techniques have made better on diffusion, latency, and security. DES is not suggested to be implemented in the real life but it is used as a benchmark to underline the efficiency of the hybrid cryptography model.

3.4.3 ASCON (Lightweight Authenticated Encryption)

ASCON has been selected as the lightweight algorithm to be used as the primary cryptography one due to its creation to operate under constrained conditions of the IoT (Radhakrishnan et al., 2024). It offers authenticated encryption, which offers confidentiality and integrity of data at minimal computational and memory expenses. The main solution to real-time IoT data encryption is that ASCON has a high diffusion capability and low latency. It is also light-weight and thus it can be easily done on simulated IoT devices without consuming resources. The selection of ASCON is aligned with the purpose of the study to improve the functionality and penetration of the IoT systems simultaneously providing an adequate degree of security and taking into account the latency as one of the primary issues in case of using cloud-connected applications.

3.4.4 ChaCha20 (Stream Cipher)

ChaCha20 was chosen because it has good performance and effectiveness in software-based implementations (Liu et al., 2025). Unlike the conventional block ciphers, ChaCha20 is a stream cipher, which makes its usage very convenient in the environment where low latency and high-speed needed encryption is required. It is reliable in other hardware platforms as well as timing resistant. In this paper, a lightweight substitute of block ciphers, ChaCha20, is evaluated and provides the possibility to compare its performance with that of ASCON. Its assistance helps to analyze throughput, latency and diffusion characteristics of cryptographic systems of hybrid IoT-cloud systems.

3.4 Performance Evaluation

3.5.1 Latency

Latency is applied to estimate the total time taken to encrypt the data at the IoT layer and transmit to cloud and receive access to cloud-based cryptographic services (Ahmad and Mehfuz, 2024). In this study, the latency is used to measure the responsiveness of the proposed hybrid cryptographic system with respect to the size of the various data and the workload. The timestamp-based measurements of the first and the final point of each cryptographic operation are the computing of processing delays. Understanding of reduced latency implies that it can be applied in real-time applications of IoT. The latency of different algorithms and configurations can be compared using this metric to determine the extent to which the lightweight encryption and parallel processing on a cloud can be utilized to reduce end-to-end delays.

3.5.2 Throughput

Throughput is the rate of successful encryption and processing data per unit time, and is normally expressed in bytes per second (Kumar et al., 2023). The throughput is applied in the current study to determine the efficiency of the system to provide the capability to address

growing volumes of data without a reduction in performance. It is determined by dividing the data size by the sum of the encryption or processing time. Increased throughput will mean improved scalability and performance especially in the clouds that have to manage multiple IoT data streams. The throughput analysis allows comparison between independent cryptographic systems and the suggested hybrid solution and shows the advantages of parallel processing and optimal workflow of cryptography.

3.5.3 CPU Utilization

CPU utilization is used to determine the amount of computation resources (Cryptographic operations) used by the devices in the IoT simulation and the cloud resources. This paper presents a monitoring of CPU usage on encryption, decryption and re- encryption processes to consider the efficiency of the resources. The metric is particularly important to IoT settings, which have devices with little processing capabilities. Reduced computer power usage in the IoT layer implies that lightweight cryptographic algorithms are appropriate. Usage of CPU at the cloud layer indicates efficiency of parallel processing and allocation of workloads. This analysis will be useful in assessing the general computational viability of the hybrid cryptographic system.

3.5.4 Avalanche Effect (Diffusion Quality)

Avalanche effect is used to assess the strength of a cryptographic algorithm in diffusion by determining the amount of alteration in input that causes an output change. A one-bit alteration of the plaintext is presented in this work and the effect of the changes on the ciphertext measured. The larger the proportion of flipped bits of output, the more it becomes diffused and the less it will withstand cryptanalytic attacks. This measure is critical in determining the quality of security of lightweight and cloud-based algorithms. The analysis of avalanche effect enables comparing the diffusion performance of ASCON, ChaCha20 and DES in the hybrid framework.

3.5.5 Encryption and Decryption Time

Encryption and decryption time is a measure of the amount of time it takes to encode plaintext into ciphertext and back to the original data respectively. These metrics are measured separately to develop the computational efficiency of every cryptographic algorithm. A shorter encryption and decryption speed is a good sign of better performance, especially in the use of latency-sensitive IoT applications. To determine timing measurements in this research, high-resolution clocks are used to take measurements. A comparison of these values in the context of various algorithms would give an idea of their applicability to IoT and cloud implementation, which will help to assess the proposed hybrid cryptographic solution.

4 Design Specification

In this chapter, the general architectural view of the proposed hybrid lightweight cryptographic framework is given and the communication between the IoT devices and the clouds is shown.

It also introduces the algorithm of the proposed model, which is the step-by-step encryption, transmission and processing process in the IoT-cloud environment.

4.1 System Architecture and Components

Figure 1 shows the overall workflow and key components of the proposed hybrid cryptographic framework deployed in an IoT–cloud environment. The system starts with the IoT/User layer where data has generated by lightweight devices is transmitted to the cloud through the internet. Within the AWS Cloud infrastructure incoming data is first handled by cloud services (EC2/Elastic Beanstalk) which manage secure communication and resource provisioning. The Message Processing module prepares the data by formatting and splitting it into suitable blocks for efficient parallel execution.

The core of the architecture is the Parallel Encryption Processing unit where two hybrid encryption schemes ASCON–DES and ASCON–ChaCha20 which have been executed concurrently to evaluate their performance. This parallel design improves scalability and enables fair comparison under identical conditions. Encrypted data has then forwarded to the results module which records performance metrics like encryption/decryption time, avalanche effect, CPU utilization, throughput and memory usage. The architecture supports secure, powerful, and scalable cryptographic evaluation used for IoT–cloud applications.

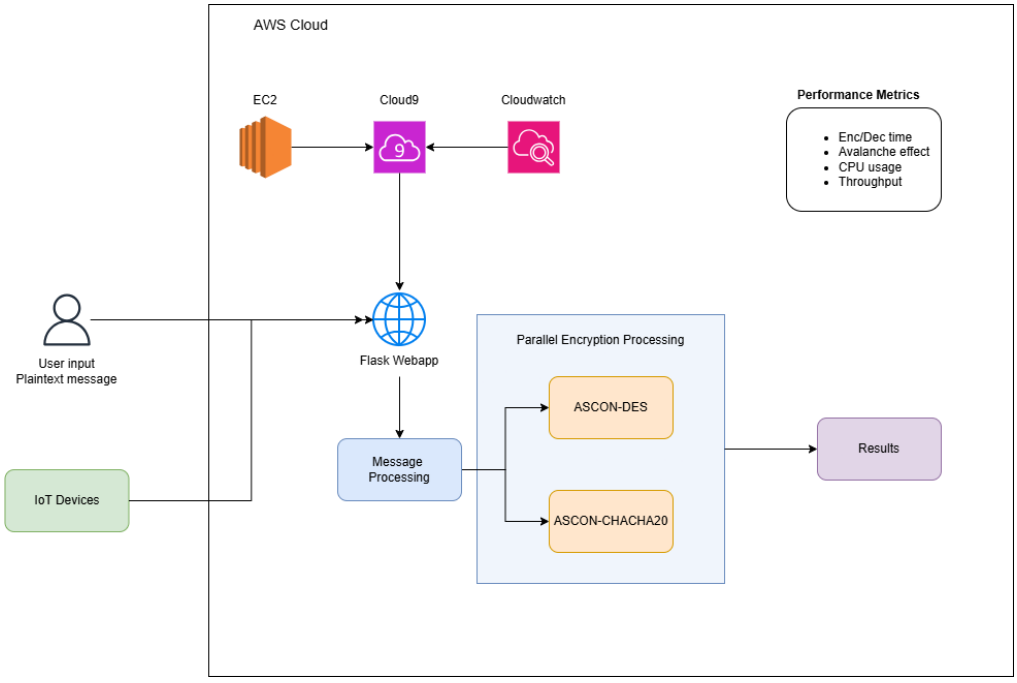

Figure 1: System Architecture Diagram

4.2 Algorithm of the Proposed Model

Algorithm: Hybrid Parallel Lightweight Cryptographic Model (ASCON + ChaCha20)

Input:
D → IoT sensor data
K1 → ASCON secret key
K2 → ChaCha20 secret key

Output:
C_final → Secure encrypted ciphertext
M → Performance metrics (latency, CPU usage, throughput)

Algorithm Steps:

1. **Start**
2. Read input data D from IoT device
3. **If** data size |D| is small
   Encrypt D using **ASCON** with key K1
   Set C1 ← ASCON_Encrypt(D, K1)
   Else
   Split D into blocks $D_1 \ldots D_n$
4. **For** each block D_i in parallel
   Encrypt D_i using **ASCON**
   Store encrypted block in $C1_i$
5. Merge all encrypted blocks into C1
6. Send C1 to cloud server
7. Cloud server encrypts C1 using **ChaCha20** with key K2
   Set C_final ← ChaCha20_Encrypt(C1, K2)
8. Record performance metrics M (latency, CPU, throughput)
9. **Return** C_final, M
10. **End**

5 Implementation

The chapter has provided the practical implementation of the proposed hybrid lightweight cryptographic system, which is outlined by describing the tools, technologies, and pipeline of deployment. It describes how the IoT and cloud layers are designed and implemented, integrating parallel processing, and using AWS and GitHub Actions to allow deployment and testing of systems in a scalable and automated way, as well as to make them reproducible.

5.1 Tools and Technologies Used

This table 2 summarizes the tools and technologies which is used to design, implement and evaluate the proposed hybrid cryptographic system.

Table 2: Summary of Tools and Technologies Used

Category	Tool / Technology	Purpose in the Study
Programming Language	Python 3	Development of IoT simulator, cryptographic modules, and performance measurement
Cryptographic Libraries	PyCryptodome, Cryptography	Implementation of DES, ASCON, and ChaCha20 algorithms
Lightweight Cryptography	ASCON, ChaCha20	Encryption at the IoT layer for constrained environments
Cloud Infrastructure	AWS EC2	Hosting cloud-based cryptographic services
Application Deployment	AWS Elastic Beanstalk	Deployment and management of cloud applications
Development Environment	AWS Cloud9	Cloud-based development and testing
Parallel Processing	Python Multithreading / Multiprocessing	Parallel encryption and decryption in the cloud
Version Control	GitHub	Source code management
Pipeline	GitHub Actions	Automated build and deployment

5.2 IoT Layer Design

The implementation of the IoT layer is on a Python client-side simulator that emulates resource-constrained devices in the IoT. It is meant to create synthetic sensor data and use lightweight cryptography algorithms like ASCON and ChaCha20. These algorithms were chosen because they have minimal computational requirements and they are appropriate in a constrained environment. The simulator will do encryption locally and note the time of execution and CPU consumption and send encrypted data to the cloud. This design means that sensitive data can be secured at the source but with minimum processing being done on the devices of the IoT system, hence saving on energy use and lowering the latency used during initial transmission.

5.3 Cloud Layer Design

AWS EC2 and Elastic Beanstalk are used to implement the cloud layer that has to process computationally intensive cryptographic operations. After the IoT layer transmits encrypted data to the cloud services, it is re-encrypted and decrypted by the cloud services with more secure algorithms based on the setting of the experiment. This layer deals with security, scaling and performance monitoring. The cloud-based implementation enables the system to take advantage of the high processing capabilities and memory, which ensures the safe processing of large amounts of data. This is where performance metrics are recorded in the form of latency, throughput, and CPU utilization which are evaluated.

5.4 Parallel Processing Integration

The cloud layer is improved by using Python multithreading and multiprocessing methods, which implies parallel processing to improve the performance. Encrypted data is broken into smaller blocks which are handled by a number of threads or processes simultaneously. Encryption or decryption is done after which the processed blocks are reassembled to give the final output. The method saves time on processing and enhances system throughput, especially when there are large loads of data. The proposed parallel cryptographic architecture can be scaled and its efficiency can be supported through the results of performance enhancements obtained by comparing the values of the latency and throughput to the sequential execution which confirms the effectiveness of the proposed architecture.

5.5 Deployment Pipeline (GitHub Actions + AWS)

5.5.1 AWS EC2- Cloud Cryptographic Processing Service

Amazon EC2 is the basic cloud service to deploy the cryptographic processing logic in the proposed hybrid system (Calderoni et al., 2022). In this study EC2 instances are fed with encrypted data being sent by the simulation IoT layer and carry out computationally expensive cryptographic operations like re-encryption and decryption through algorithms like ChaCha20. EC2 environment is powerful enough to support parallel cryptographic workloads in terms of processing power and memory. Encryption APIs written in Python are put to run on EC2 and to process incoming requests and give out encrypted or decrypted responses. Also, EC2 records the amount of time taken, CPU utilization and throughput, which is subsequently used to evaluate the performance. Scaling of computational resources is achieved using EC2; therefore, it is appropriate to use EC2 in the analysis of cloud-based cryptography between peaks and slack workloads.

5.5.2 AWS Elastic Beanstalk -Application Deployment and Management

The cloud-based cryptographic application is deployed and managed with AWS Elastic Beanstalk to make the process easier. Elastic Beanstalk is used in this project where the Python encryption APIs are deployed on EC2 and do not need a manual configuration of the infrastructure components. It takes care of deployment of applications, provisioning of the environment, load balancing, and monitoring automatically. With this service, it is possible to update the cryptographic logic quickly and make sure that the app is available throughout the testing. Version management is also provided by Elastic Beanstalk, which enables the implementation of various experimental settings and test them separately. Elastic Beanstalk removes cloud management complexity which enables the research to concentrate on cryptographic performance and analysis instead of cloud management which makes it an ideal fit to the deployment of scalable cryptographic service in a controlled experimental setting.

5.5.3 AWS Cloud9 Development and Testing Environment

The cryptographic system is implemented and tested with the help of AWS Cloud9 which serves as the cloud-based integrated development environment (IDE). Cloud9 is used in this study to create and test Python scripts to simulate, encrypt, decrypt, and log the performance

of IoT components. It allows direct access to AWS resources like EC2 and Elastic Beanstalk, and therefore, cloud services can be debugged and tested in real-time. Cloud9 facilitates multi-user development, and provides consistency between development environments between experiments. Cryptography operates can be directly tested in the AWS ecosystem and this decreases configuration errors and compatibility in the development and deployment phases, which plays a role in efficient and reliable implementation of systems.

5.5.4 GitHub Actions Pipeline

The study is implemented with a pipeline implemented with the help of GitHub Actions. It automates the creation, testing as well as deployment of cryptographic applications on AWS services. Each time the code is pushed to the GitHub repository, GitHub Actions causes workflows, which check the Python code, run test scripts, and deploy new versions in Elastic Beanstalk and EC2. The consistency of deployments and minimized manual intervention is achieved through this automation. GitHub Actions is also helpful in fast experimentation through the possibility of several configurations of cryptographic algorithms deployed on GH Actions. The pipeline increases the reliability, reproducibility and efficiency in testing the hybrid cryptographic system under different experimental conditions.

6 Evaluation

This chapter presents a performance evaluation of the proposed hybrid encryption schemes by experimentally analyzing encryption–decryption time, latency, throughput, CPU and memory utilization and avalanche effect. The results has showed that the Hybrid ASCON–ChaCha20 approach consistently outperforms Hybrid ASCON–DES in terms of performance, scalability, and diffusion strength by making it more suitable for secure IoT–cloud environments.

6.1 Experiment 1: Encryption and Decryption Time

The current experiment compares the time of encryption and decryption of two hybrid cryptography methods i.e. Hybrid ASCON-DES and Hybrid ASCON-ChaCha20 in a parallel execution system. The findings in Table 3 show that the Hybrid ASCON-ChaCha20 method incurs a lesser encryption and decryption time than the Hybrid ASCON-DES. This has been made possible mainly due to the stream-based architecture of ChaCha20, which is capable of generating key stream faster and can be readily parallelized.

Whereas DES uses several rounds and adds to computation costs both encryption and decryption. The hybrid architecture employs the ASCON at the lightweight IoT level, and it offloads the heavy processing to the cloud, thus making it time-efficient. The implementation process, however, graphically experienced the difficulty of overheads in synchronization and safe management of key across parallel threads. Regardless of these complexities, Hybrid ASCON-ChaCha20 performs better based on lower complexity of the algorithm and maximized diffusion rendering it even more appropriate in time-sensitive IoT cloud applications.

Table 3: Encryption and Decryption Time Comparison of Hybrid Cryptographic Techniques

Algorithm	Encryption Time (s)	Decryption Time (s)
Hybrid ASCON–DES	1.556	1.115
Hybrid ASCON–ChaCha20	1.391	0.913

6.2 Experiment 2: Latency and Throughput

In this experiment, system level performance is examined in terms of latency and throughput of both hybrid encryption schemes as shown in Table 4. The end-to-end processing time is the delay of time taken and the amount of data taken/processed per unit time is the throughput. The findings indicate that Hybrid ASCON-ChaCha20 has higher throughput than Hybrid ASCON–DES, which has slightly low latency. This low latency in ASCON- DES would be attributed to its simpler block-based processing in parallel implementation.

Nevertheless, ChaCha20 is much more effective in improving throughput because it is capable of handling data streams in real-time, without using frequent synchronization barriers. High throughput also presented problems of high CPU usage and overhead of scheduling of tasks in parallel execution. However, the Hybrid ASCON-ChaCha20 solution is more scalable and has higher enduring data processing rates. As a result, ASCON-DES can be used in the operations that need the latency, whereas ASCON-ChaCha20 is better applicable in the high-throughput IoT-cloud systems that need the continuous data transmission.

Table 4: Latency and Throughput Analysis of Hybrid Encryption Schemes

Algorithm	Latency	Throughput
Hybrid ASCON–DES	2.915	28.87
Hybrid ASCON–ChaCha20	2.974	32.3

6.3 Experiment 3: CPU and Memory Utilisation Analysis

This experiment analyzes the computational overhead of the proposed hybrid encryption schemes by evaluating CPU utilization and memory consumption as shown in Table 5. CPU usage reflects the processing load which has been imposed by cryptographic operations, while memory usage indicates the space required for algorithm execution and data handling. The results show that Hybrid ASCON–DES incurs higher CPU utilization (89.36%) due to the computational complexity of DES rounds combined with parallel encryption overhead. Whereas Hybrid ASCON–ChaCha20 demonstrates negligible CPU usage showing its lightweight and computation-efficient design.

Memory consumption for both approaches remains nearly identical by showing that neither algorithm introduces substantial memory overhead. Although achieving low CPU utilization required careful optimization of parallel processing and thread management, Hybrid ASCON–

ChaCha20 clearly outperforms ASCON–DES in terms of computational performance. These results confirm the suitability of ASCON–ChaCha20 for deployment in IoT–cloud systems with limited processing resources.

Table 5: CPU and Memory Analysis of Hybrid Encryption Schemes

Algorithm	CPU	Memory
Hybrid ASCON–DES	89.36	41.422
Hybrid ASCON–ChaCha20	0	41.414

6.4 Experiment 4: Avalanche and Diffusion Analysis

The experiment determined the diffusion strength of the proposed hybrid encryption schemes on the basis of avalanche effect as shown in Table 6. The avalanche effect is used to determine the proportion of ciphertext bits that change with a small change to the plaintext and a larger value implies that the diffusion is more effective and that the encryption is more resistant to differential cryptanalysis. The findings indicate that the Hybrid ASCON-ChaCha20 method has much higher average avalanche score of 99.956% than 94.054% of Hybrid ASCON-ChaCha20.

This is enhanced mainly by ChaCha20 stream based architecture and the ability to add operations in a modular fashion which permits fast and pervasive propagation of inputs on the ciphertext. Conversely, DES uses predetermined Feistel rounds and this restricts the power of diffusion when their weight is light. In the hybrid parallel structure, synchronization of avalanche scores had to be done carefully and the key scheduling had to be secure. Generally, the best performance demonstrated by Hybrid ASCON-ChaCha20 in avalanches is the sure sign of its better diffusion capacity, which makes it a more secure and robust option in terms of applying in IoT-cloud cryptography.

Table 6: Avalanche Effect Comparison

Algorithm	Avalanche Score (%)
Hybrid ASCON–DES	94.054
Hybrid ASCON–ChaCha20	99.956

6.5 Experiment 5: Statistical Performance Evaluation of Hybrid Lightweight Cryptographic Schemes

The statistical evaluation of both hybrid cryptographic schemes shows clear performance differences arising from their algorithmic design and execution characteristics. The ASCON–ChaCha20 framework consistently achieves better performance due to ChaCha20's stream-based encryption which enables faster processing, efficient parallel execution, and smoother handling of continuous IoT data streams in cloud environments. Whereas the ASCON–DES

scheme introduces additional computational overhead and synchronization costs during parallel processing. The ASCON–ChaCha20 hybrid framework is the superior solution due to its powerful stream-based encryption design which supports faster execution and good parallelization in cloud environments

Table 7: Statistical Performance Results of ASCON–DES Hybrid Cryptographic Scheme

Metric	Mean	Std. Deviation	Minimum	Maximum	95% Confidence Interval (±)
Encryption Time (ms)	2.2311	1.1338	1.028	8.540	0.2485
Decryption Time (ms)	2.1253	1.1830	0.946	6.110	0.2592
Latency (ms)	4.6507	1.7880	2.128	9.874	0.3918
Throughput (KB/s)	13.4668	9.3522	0.91	34.38	2.0494
CPU Usage (%)	11.1462	28.7139	0	100	6.2922
Memory Usage (MB)	21.7966	0.0163	21.719	21.828	0.0036

Table 8: Statistical Performance Results of ASCON–ChaCha20 Hybrid Cryptographic Scheme

Metric	Mean	Std. Deviation	Minimum	Maximum	95% Confidence Interval (±)
Encryption Time (ms)	1.5958	0.6407	0.846	3.414	0.1404
Decryption Time (ms)	1.5077	0.9345	0.849	7.249	0.2048
Latency (ms)	3.3845	1.3726	1.849	8.834	0.3008
Throughput (KB/s)	17.2214	11.1030	2.29	43.52	2.4330
CPU Usage (%)	9.6863	25.8270	0	100	5.6596
Memory Usage (MB)	21.7054	0.0224	21.637	21.754	0.0049

7 Conclusion and Future Work

This study has explored how an ASCON-ChaCha20-based hybrid lightweight cryptographic system with parallel cloud-based processing can be used to improve the security of IoT-related data more than the traditional cloud encryption systems. The experimental evidence indicates

that the suggested hybrid technique can considerably lower the encryption time and decryption time and higher throughput when using concurrent workloads. ASCON at the IoT level and ChaCha20 at the cloud level are lightweight and energy-efficient, and secure and fast stream-based processing, respectively. Parallel running on the cloud also enhances performance through effective utilization of CPU resources which increase diffusion properties and the responsiveness of the entire system. These results address the research question which directly establish that hybrid lightweight cryptography with cloud parallelism have measurable performance and diffusion benefits over traditional methods.

These contributions are accompanied by some limitations of the study. This test was done with the help of an artificially created controlled cloud scenario with simulated IoT devices; it may not be a good representation of the variability of networks and hardware constraints in practice. Also security analysis was performed on performance based measures, such as diffusion and avalanche effects, rather than formal cryptanalytic resistance.

Future directions will further develop such a framework to the actual physical implementation of the IoT, application after the quantum lightweight cryptographic constructions, and rigorously test the energy usage. Further optimization such as adaptive autoscaling and real-time threat-sensitive encryption choice can also be implemented to make the system offered more resilient and increasingly feasible.

References

Abba Ari, A.A., Ngangmo, O.K., Titouna, C., Thiare, O., Kolyang, Mohamadou, A., Gueroui, A.M., 2024. Enabling privacy and security in Cloud of Things: Architecture, applications, security & privacy challenges. Appl. Comput. Inform. 20, 119–141. https://doi.org/10.1016/j.aci.2019.11.005

Ahmad, S., Mehfuz, S., 2024. Efficient time-oriented latency-based secure data encryption for cloud storage. Cyber Secur. Appl. 2, 100027. https://doi.org/10.1016/j.csa.2023.100027

Ahmad, S., Mehfuz, S., Beg, J., 2023. Hybrid cryptographic approach to enhance the mode of key management system in cloud environment. J. Supercomput. 79, 7377–7413. https://doi.org/10.1007/s11227-022-04964-9

Almutairi, M., Sheldon, F.T., 2025. IoT–Cloud Integration Security: A Survey of Challenges, Solutions, and Directions. Electronics 14, 1394. https://doi.org/10.3390/electronics14071394

Barriga, J.A., Clemente, P.J., Hernandez, J., Perez-Toledano, M.A., 2022. SimulateIoT-FIWARE: Domain Specific Language to Design, Code Generation and Execute IoT Simulation Environments on FIWARE. IEEE Access 10, 7800–7822. https://doi.org/10.1109/ACCESS.2022.3142894

Biswas, S., Raj, Md.W.Z., Master of science in Information Technology Management, Cumberland University, Tennessee, USA, 2024. QUANTUM-RESISTANT CRYPTOGRAPHIC PROTOCOLS INTEGRATED WITH AI FOR SECURING CLOUD AND IOT ENVIRONMENTS. Int. J. Bus. Econ. Insights 04, 60–90. https://doi.org/10.63125/dryw3b96

Calderoni, L., Maio, D., Tullini, L., 2022. Benchmarking Cloud Providers on Serverless IoT Back-End Infrastructures. IEEE Internet Things J. 9, 15255–15269. https://doi.org/10.1109/JIOT.2022.3147860

Dhanda, S.S., Singh, B., Jindal, P., 2020. Lightweight Cryptography: A Solution to Secure IoT. Wirel. Pers. Commun. 112, 1947–1980. https://doi.org/10.1007/s11277-020-07134-3

Emre Erbil, M., Cenk Bayrakci, H., Özkahraman, M., 2025. A NOVEL HYBRID ENCRYPTION SCHEME BASED ON MODERN CRYPTOGRAPHIC COMPONENTS FOR EMBEDDED SYSTEMS.

Farooq, S., Chawla, P., Kumar, N., 2023. A cryptographic security framework for hybrid CLOUD-INTERNET OF THINGS network. Secur. Priv. 6, e309. https://doi.org/10.1002/spy2.309

Farshadinia, H., Barati, A., Barati, H., 2025. A secure and energy-efficient architecture in Internet of Things–cloud computing network by enhancing and combining three cryptographic techniques via defining new features, areas, and entities. J. Supercomput. 81, 944. https://doi.org/10.1007/s11227-025-07390-9

Javadpour, A., Ja'fari, F., Taleb, T., Zhao, Y., Yang, B., Benzaïd, C., 2024. Encryption as a Service for IoT: Opportunities, Challenges, and Solutions. IEEE Internet Things J. 11, 7525–7558. https://doi.org/10.1109/JIOT.2023.3341875

Justindhas, Y., Jeyanthi, P., 2023. Secured Model for Internet of Things (IoT) to Monitor Smart Field Data with Integrated Real-Time Cloud Using Lightweight Cryptography. IETE J. Res. 69, 5134–5147. https://doi.org/10.1080/03772063.2021.1966844

Khan, M.N., Rao, A., Camtepe, S., 2021. Lightweight Cryptographic Protocols for IoT-Constrained Devices: A Survey. IEEE Internet Things J. 8, 4132–4156. https://doi.org/10.1109/JIOT.2020.3026493

Kumar, A., Singh, P., Patro, K.A.K., Acharya, B., 2023. High-throughput and area-efficient architectures for image encryption using PRINCE cipher. Integration 90, 224–235. https://doi.org/10.1016/j.vlsi.2023.01.011

Liu, C., Zhao, S., Jia, C., Hu, G., Cui, T., 2025. An Improved ChaCha Algorithm Based on Quantum Random Number. https://doi.org/10.48550/ARXIV.2507.18157

Mishra, S., Tyagi, A.K., 2022. The Role of Machine Learning Techniques in Internet of Things-Based Cloud Applications, in: Pal, S., De, D., Buyya, R. (Eds.), Artificial Intelligence-Based Internet of Things Systems, Internet of Things. Springer International Publishing, Cham, pp. 105–135. https://doi.org/10.1007/978-3-030-87059-1_4

Muhammed, R.K., Rashid, Z.N., Saydah, S.J., 2025. A Hybrid Approach to Cloud Data Security Using ChaCha20 and ECDH for Secure Encryption and Key Exchange. Kurd. J. Appl. Res. 10, 66–82. https://doi.org/10.24017/science.2025.1.5

Patel, A., Cherukuri, A.K., 2025. Analysis of Light-Weight Cryptography Algorithms for UAV-Networks. https://doi.org/10.48550/ARXIV.2504.04063

Pothireddy, S., Peddisetty, N., Yellamma, P., 2024. Data Security in Cloud Environment by Using Hybrid Encryption Technique: A Comprehensive Study on Enhancing Confidentiality and Reliability. Int. J. Intell. Eng. Syst. 17, 159–170. https://doi.org/10.22266/ijies2024.0430.14

Potlapally, N.R., Ravi, S., Raghunathan, A., Jha, N.K., 2006. A study of the energy consumption characteristics of cryptographic algorithms and security protocols. IEEE Trans. Mob. Comput. 5, 128–143. https://doi.org/10.1109/TMC.2006.16

Qasem, M.A., Thabit, F., Can, O., Naji, E., Alkhzaimi, H.A., Patil, P.R., Thorat, S.B., 2024. Cryptography algorithms for improving the security of cloud-based internet of things. Secur. Priv. 7, e378. https://doi.org/10.1002/spy2.378

Radhakrishnan, I., Jadon, S., Honnavalli, P.B., 2024. Efficiency and Security Evaluation of Lightweight Cryptographic Algorithms for Resource-Constrained IoT Devices. Sensors 24, 4008. https://doi.org/10.3390/s24124008

Ramakrishna, D., Ali Shaik, M., 2025. A Comprehensive Analysis of Cryptographic Algorithms: Evaluating Security, Efficiency, and Future Challenges. IEEE Access 13, 11576–11593. https://doi.org/10.1109/ACCESS.2024.3518533

Ranganatha Rao, B., Sujatha, B., 2023. A hybrid elliptic curve cryptography (HECC) technique for fast encryption of data for public cloud security. Meas. Sens. 29, 100870. https://doi.org/10.1016/j.measen.2023.100870

Singha, T.B., Palathinkal, R.P., Ahamed, S.R., 2023. Securing AES Designs Against Power Analysis Attacks: A Survey. IEEE Internet Things J. 10, 14332–14356. https://doi.org/10.1109/JIOT.2023.3265683