National College of Ireland

A Hybrid ChaCha20-Based Lightweight Cryptographic Framework for Latency and Diffusion Optimization in IoT Systems

MSc Research Project
Programme Name

Nishanth Marisamy
Student ID: X23331968

School of Computing
National College of Ireland

Supervisor: Eugene Mclaughlin

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Nishanth Marisamy
Student ID:	X23331968
Programme:	MSc In Cyber Security **Year:** 2025-2026
Module:	MSc In Cyber Security
Supervisor:	Eugene Mclaughlin
Submission Due Date:	11/12/2025
Project Title:	A Hybrid ChaCha20-Based Lightweight Cryptographic Framework for Latency and Diffusion Optimization in IoT Systems
Word Count:	8,882 **Page Count** : 24

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: Nishanth Marisamy

Date: 11/12/2025

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 ChaCha20-Based Lightweight Cryptographic Framework for Latency and Diffusion Optimization in IoT Systems

Nishanth Marisamy
X23331968

Abstract

The rapid expansion of IoT ecosystems has increased the need for secure yet low-latency cryptographic solutions capable of operating on constrained devices. There are some existing studies which attempts to optimize AES or ChaCha20 for lightweight environments fail to fully address the combined challenges of high latency, weak diffusion and limited scalability in cloud-integrated IoT systems. This study was undertaken to solve this gap by developing a hybrid cryptographic architecture that leverages lightweight ciphers (SPECK, SIMON) for fast edge-level processing and robust ciphers (AES or ChaCha20) for strong diffusion and enhanced security. A clear threat model has also defined to evaluate the system under some realistic IoT security conditions which includes active network adversaries, replay attempts and chosen-plaintext attacks. The proposed system contains parallel processing, block-level optimization and a layered encryption pipeline to achieve powerful balance between performance and protection. Experimental evaluation shows some important improvements with diffusion scores reaching up to 97% and noticeable reductions in encryption and decryption latency compared to standalone AES and ChaCha20 systems. These results show that the hybrid model advances the state of the art by integrating multi-layer encryption with IoT-oriented performance giving stronger unpredictability and lower computational overhead. This study investigates whether a hybrid lightweight cryptographic model combining SPECK, SIMON and ChaCha20 can improve latency and diffusion as compared to individual ChaCha20 in IoT environment.

Keywords: IoT Cybersecurity, Lightweight Cryptography, ChaCha20, Hybrid Encryption

1 Introduction

This chapter outlines the background of cybersecurity and IoT, presents the problem statement, states the research question, defines the research objectives and provides the overall structure of the study, explaining how each section contributes to developing and evaluating the proposed hybrid cryptographic system.

1.1 Background

The rapid digital transformation in all industries has made cybersecurity a crucial issue on the global agenda (Möller, 2023) especially as the systems are becoming more and more interconnected via cloud computing and the Internet of Things (IoT). Current IoT environments are based on the idea of constant communication (Dritsas and Trigka, 2025) between sensors, devices and cloud platforms to form massive distributed environments that process sensitive data in real time (Ficili et al., 2025). Nonetheless, this interconnectivity increases the area of attack, placing IoT systems at risk as an interceptor of data, unauthorized entry, device abuse and massive distributed attacks. The cryptographic algorithms such as AES are known to be computationally intensive and hence less practical in Layered Lightweight Encryption (LLE) IoT devices with low processing capability, battery capacity and memory, although they are secure. Meanwhile, numerous lightweight ciphers (Mohammad and Abdullah, 2022) are often faster and efficient, but are potentially lacking strong diffusion or resistance to failure that is needed with cloud-integrated use. With the development of cyber threats that are more sophisticated, it becomes more difficult to find the appropriate balance between security level and performance. This has led to a move towards hybrid cryptography designs that use the strength of conventional ciphers, but with the efficiency of lightweight algorithms. Parallel processing, edge cloud coordination and algorithmic optimization also help in the improvement of security and in addition, low latency is maintained (Sathish kumar, 2023). In that regard, hybrid lightweight encryption methods have proven that they are heading towards a promising path of delivering secure, efficient and scalable IoT deployments.

1.2 Problem Statement

The rapid growth of Internet of Things (IoT) devices and their integration with cloud computing presents a great challenge in terms of data security (Nag et al., 2024) particularly when operating with high constraints of low power, limited processing power, and real-time response. Traditional cryptographic algorithms like AES are very secure but computationally expensive meaning they have a higher latency and less efficiency on low-weight machines. Lightweight ciphers, such as SPECK and SIMON, on the other hand, are adapted to constrained settings but possibly not have strong diffusion or can be interoperable with a cloud implementation.

Current solutions that seek to adapt ChaCha20 to a lightweight application do not comprehensively consider the security strength-latency-diffusion quality trade-off, especially in a parallelizable, cloud-integrated IoT scenario.

The study will focus on establishing a hybrid cryptography system using lightweight and traditional cryptographic systems, in a parallelized cloud computing platform, in order to improve security, minimize latency, and maximize diffusion with IoT systems.

1.3 Research Question

How can a hybrid cryptographic approach using Layered Lightweight Encryption (LLE) techniques improve latency and diffusion performance in IoT environments compared to standalone ChaCha20 systems?

1.4 Research Objectives

1. To design and implement a hybrid cryptographic framework that integrates ChaCha20 with lightweight ciphers such as SPECK and SIMON to reduce encryption latency and enhance diffusion performance in IoT systems.
2. To evaluate the hybrid model against standalone ChaCha20 and AES implementations using metrics such as latency, throughput, CPU usage, encryption/decryption time, and avalanche effect.
3. To assess the effectiveness of parallel processing within the hybrid cryptographic system in optimizing performance for cloud-integrated IoT environments with constrained computational resources.

1.5 Structure of the Report

This report is organized into six chapters. Chapter 1 introduces the background, problem statement, research question, objectives, and report structure. Chapter 2 presents related work on lightweight, traditional, and hybrid cryptography, including gaps in existing studies. Chapter 3 explains the research design, simulation environment, diffusion analysis, and algorithm selection. Chapter 4 outlines the system architecture, IoT encryption, and decryption processes. Chapter 5 details the full implementation environment, lightweight and robust cipher execution, hybrid framework, and user interface. Chapter 6 provides evaluation results from all experiments and a comparative analysis with the baseline study. Chapter 7 shows conclusion and future work.

2 Related Work

This chapter reviews existing research on lightweight cryptography for IoT, limitations of traditional ciphers like AES and ChaCha20, and various hybrid cryptographic models. It analyses previous studies, techniques, performance results, and limitations, highlighting the need for balanced, efficient, and secure hybrid encryption approaches suitable for constrained IoT and cloud-integrated environments.

2.1 Lightweight Cryptography in IoT Systems

The Internet of Things (IoT) ecosystems have grown in rapid speed increasing the pressure on cryptographic solutions that can offer a high level of security and can be effectively run on resource-limited hardware (Mansoor et al., 2025). The Lightweight cryptography is able to

accommodate this requirement by providing algorithms to devices with limited memory, incapacity to perform calculations and limited battery power (Singh et al., 2024). Specially designed ciphers like SPECK and SIMON are used in very limited contexts, and they use round simplified structures and other operations to generate high-speed and low-computational-cost ciphers. The small size of the IoT nodes enables it to encrypt with a shorter latency, better throughput, and less energy usage than the conventional ciphers.

Although lightweight algorithms have these benefits, studies have always pointed out inherent weaknesses (Chen et al., 2024). Most lightweight ciphers have lesser diffusion properties, that is, even minor alterations in the plaintext do not extend effectively through the ciphertext, thus attenuating the avalanche effect. Such inefficient diffusion makes them vulnerable to a differential attack and can reveal the pattern of the encrypted communication. Also, lightweight ciphers tend to count reduced rounds to maintain their efficiency, which may further reduce resistance to cryptanalytic attacks (Bhagat et al., 2023). These vulnerabilities increase as IoT systems become more complex resulting in the need to develop hybrid cryptography, which balances high-speed lightweight ciphers with high diffusion and strong resistance of traditional algorithms.

2.2 Traditional High-Security Ciphers for IoT: Limitations of AES and ChaCha20

Traditional high-security ciphers like AES and ChaCha20 are commonly known to have good cryptographic features and are resistant to sophisticated attacks. AES, which is based on a substitution permutation network, has very good diffusion and avalanche effect hence is used in secure communications (Dhall and Yadav, 2024). A stream cipher called ChaCha20 is based on ARX and it is more flexible and resistant to timing attacks and it has consistent performance (Dhall and Yadav, 2024) of platform. Nonetheless, even at the back of their virtues, both of the algorithms have prominent disadvantages when used within an IoT setup where there are no substantial computational and power resources.

AES is a nonlinear transformation that needs numerous nonlinear transformations and has a significant memory requirement (Singha et al., 2023), which is computationally expensive on microcontrollers without hardware acceleration. This results in higher encryption times, more power usage and an increase in latency- none of which is appropriate in real time IoT systems. Whereas ChaCha20 is a faster algorithm than AES (ElRashidy et al., 2024) on a limited CPU, nonetheless, it also takes multiple rounds to reach the optimal level of diffusion, which also leads to visible delays when using small packet transmissions that occur frequently in an IoT network. Altered lightweight implementations of ChaCha 20 suggested in literature usually undermine either security or consistency in its performance. Therefore, both AES and ChaCha20 are highly protective, but due to the computational load, neither can be used on edge-level IoT computing devices, and hybrid architectures, where heavy encryption is farmed to the cloud and low-latency transportation maintained at the device layer, are suggested.

2.3 Hybrid Cryptographic Models for IoT Security

Extensive research on hybrid cryptographic schemes has been conducted to deal with the increasing security needs in the IoT, health, fog-cloud systems, and constrained resources. First study given by (Ali and Anwer, 2024a) that offered an authentication based model of securing the transmission of the IoT data through the combination of Elliptic Curve Cryptography (ECC), Genetic Algorithm (GA) to optimize the key generation, and AES encryption demonstrating better throughput, less execution time, and better avalanche compared to DES and RSA. (Bhandiwad and Ragha, 2025) was interested in lightweight hybrid encryption in low-power IoT gadgets, where a framework is introduced to integrate the Advanced ElGamal Signature with the AES-DES hybrid encryption to address the shortcomings of single cryptographic algorithms, with throughput of 99.68 percent and low delay, but brute-force attacks are still a problem. Along these lines of hybridization, (Sumathi M S et al., 2023) discussed the combination of symmetric, asymmetric, and hashing primitives to refine the flaws of the single cryptographic families and proposed a hybrid AI-IoT security model that produces more efficient encrypted messages, showing the value of the hybrid approach with the help of user perception data, but there are still issues of complexity and scalability. Similarly, (Patil et al., 2022) proposed a Hybrid and Adaptive Cryptographic (HAC) authentication model based on XOR operations, hashing, and dual hybrid encryption with AES-ECC and RSA-AES to enhance the authentication with minimum complexity, and claimed a low communication and memory cost, but overhead can still be a problem with ultra-low-power devices. A more system architecture is observed in (Farooq et al., 2023), where a four-phased hybrid Cloud-IoT security model combines mutual authentication, key distribution, and hybrid encryption, minimizing the encryption and decryption time by 28 and 32 percent respectively but has not been tested on real heterogeneous cloud systems. To increase the secrecy further, (Ali and Anwer, 2024b) suggested hybrid DNA-GA-ECC encryption algorithm, which enhances the efficiency of producing keys and encrypting files of different sizes but the biological and evolutionary factors add additional computational burden to devices with limited resources. Within the concepts of medical cloud security, (Gadde et al., 2023) introduced a multi-layered hybrid cryptographic scheme that integrates IRS-AES (optimized based on Runge Kutta optimization), Improved Huffman compression, DNA-based MECC, and BES-based key selection that performs well in image, audio, and video data. However, the large optimization pipeline makes the system more complex and can make it hard to deploy the system lightweight as shown in Table 1. Lastly, (Corthis et al., 2024) improved upon the security of a healthcare IoT using fog with the help of synergizing ECC and Proxy Re- Encryption (PR) with the Enhanced Salp Swarm Algorithm (ESSA) to optimize key and parameter selection to as low as 18 ms compared to 60 ms, at the cost of fog-node dependency and integration complexity. Taken together, these research works emphasize the efficiency of hybrid cryptographic systems in addressing the inherent drawbacks of symmetric, asymmetric, and bio-inspired cryptographic techniques, and continue to suggest that there are still significant obstacles to consider in terms of computational complexity, scalability, and real-life implementation in a variety of IoT systems.

Study No.	Purpose / Focus	Proposed Hybrid Approach	Key Methods / Techniques Used	Main Results	Limitations
(Ali and Anwer, 2024a)	Secure IoT authentication and confidentiality	ECC + GA-based key generation + AES encryption	User & device authentication, cloud data protection	Improved execution time, throughput, strong avalanche effect; better than DES & RSA	GA-based key generation may increase computation overhead
(Bhandiwad and Ragha, 2025)	Lightweight IoT security for constrained devices	Hybrid AES–DES + Advanced ElGamal Signature	Python implementation, adaptive encryption	High throughput (99.68%), 100% transmission rate, 0.3 s delay	Vulnerable to brute-force; complexity of dual encryption
(Sumathi M S et al., 2023)	Improve AI–IoT security using hybrid cryptography	Symmetric + Asymmetric + Hash hybrid	Comparative hybrid cryptographic evaluation	More efficient encrypted outputs; users view faster threat assessment as important (44.1%)	Scalability issues; multi-scheme integration complexity
(Patil et al., 2022)	Secure authentication for IoT systems	Hybrid & Adaptive Cryptographic (HAC): AES–ECC + RSA–AES	XOR, hashing, dual hybrid encryption	Low communication cost (0.017 s), low computation time, minimal memory use	Dual-hybrid methods may burden low-power IoT nodes
(Farooq et al., 2023)	Secure cloud–IoT communication framework	Four-phase model: mutual authentication + key distribution + hybrid encryption	Novel authentication & key distribution algorithms	28% faster encryption, 32% faster decryption, higher throughput	Limited real-world validation; cloud heterogeneity challenges
(Ali and Anwer, 2024b)	Confidentiality for IoT data	DNA Cryptography + Genetic Algorithm + ECC	Hybrid key generation & encryption pipeline	Faster key generation, encryption & decryption for varied file sizes	Bio-inspired cryptography increases computational load
(Gadde et al., 2023)	Protect isolation of medical data in cloud	IRS-AES + RKO + DNA-MECC + BES key optimization	Huffman compression, optimized S-box, multi-media evaluation	High robustness; improved communication overhead, PSNR, MSE across datasets	Highly complex layered system; heavy for resource-limited IoT devices
(Corthis et al., 2024)	Fog-based secure healthcare IoT authentication	ECC + Proxy Re-Encryption (PR) + ESSA	Fog computing, hybrid optimization (WOA + SSA)	Processing time reduced (60→18 ms), 25% reliability improvement, low memory use	Dependency on fog infrastructure; optimization overhead

Table 1: Summary Table

2.4 Security Foundations and Algorithm Critiques

The performance of hybrid encryption and composite encryption systems is based on layered encryption design concepts like confusion, diffusion and formal adversarial models like CPA (Chosen-Plaintext Attack) and CCA (Chosen-Ciphertext Attack) resistance. Based on NIST lightweight cryptography reports (NISTIR 8114 and NIST LWC draft standards), the balance between performance and resource trade-offs and provable security margins have to be established. Although NSA designed SPECK and SIMON as lightweight ciphers, these two algorithms are the subject of criticism in the literature because of the apparent lack of transparency in their design justification and the possible structural weakness of these two algorithms. Some researchers also cast doubt on their applicability in the long term security and NIST removed them off the ultimate lightweight cryptography recommendations. These criticisms point to a rational choice of ciphers in limited settings.

2.5 Gaps and Limitations in Existing Literature

Existing literature identifies the major gaps in IoT cryptography, specifically the inability of lightweight ciphers to provide strong diffusion and the high latency caused by conventional algorithms, such as AES, when applied single-handedly on limited devices. Combining several algorithms, many hybrid strategies, nonetheless, will have a high level of complexity in their calculation, poor scaling, or will not be evaluated in terms of the real IoT. Parallel processing, bit-level avalanche analysis, and balanced multi-layer encryption optimized to edge-cloud IoT conditions are also not explored in most of the studies. In addition, existing ChaCha20 lightweight implementations between security and diffusion and latency. The proposed ChaCha20 and SPECK/SIMON framework directly overcomes those limitations.

3 Research Methodology

This chapter explains the research design using CRISP-DM, details the IoT security simulation environment, describes diffusion and avalanche analysis, and outlines algorithm selection. It covers the hybrid encryption modelling process, evaluation methods, and justification for using AES+SPECK+SIMON and ChaCha20+SPECK+SIMON within the proposed cryptographic framework.

3.1 Research Design Framework

The study uses the CRISP-DM (Cross-Industry Standard Process for Data Mining) model, which has been modified to fit the context of cybersecurity-related cryptographic systems development to render the study systematic and repeatable. The paper commences with Business Understanding, which determines the nature of the fundamental issue of high latency, poor diffusion, and security constraints of the current IoT encryption models, especially when relying on the heavy algorithms such as AES or lightweight ciphers such as SPECK and SIMON, alone. The Data Understanding stage reviews IoT communication

patterns, threat surfaces, payload sizes and adversarial risks, which have an effect on cryptographic behavior. In Data Preparation, plaintext datasets and generated IoT traffic can be organised into block format to be examined in the cipher. The Modelling step is aimed at the design and implementation of the Hybrid Lightweight Encryption system wherein in the IoT, lightweight ciphers run the Hybrid Lightweight Encryption system whereas in the IoT, there is AES/ChaCha20 encryption of high strength (Jakhu, 2025). Secure key management, block partitioning and parallel processing are also involved in this phase. During the Evaluation phase, the models are evaluated in the latency tests, avalanche analysis, entropy scoring, and the attack-resistance validation. Lastly, Deployment simulates actual IoT-interactions to determine how the hybrid model enhances the confidentiality, integrity, diffusion quality, and the general security and retains efficiency in the limited setting.

3.2 IoT Security Simulation Environment

The Simulation Environment The IoT Security Simulation Environment used in this work is aimed at stimulating the behaviour of a real-world constrained IoT device (Almutairi et al., 2024) and allowing to control a hybrid lightweight encryption approach. The environment is written completely in Python, and is a client-side simulator, which models low-power device behaviour, such as limited CPU capability, small code footprint, and real-time bandwidth limitations. In this simulation, the data is created, pre-processed, and encrypted with lightweight ciphers (SPECK and SIMON) and this corresponds with the encryption processes that are normally done at the edges of the IoT networks. The simulator is modelled to perform hybrid encryption operations such that SPECK or SIMON is used to encrypt first the plaintext and then AES or ChaCha20 in a stacking fashion to test diffusion and latency enhancements. To closely replicate real world IoT functioning, parallel processing is introduced into the environment using Python Multithreading or Multi Processing module which allows efficient block-by-block encryption and decryption. This is a setup that enables proper measurement of such performance attributes as encryption time, decryption time, CPU load, throughput, avalanche effect. The environment allows the hybrid cryptographic design to be tested in a controlled way by simulating the behaviour of the IoT locally, without external dependencies so that a consistent, lightweight and controlled platform is obtained to verify the effectiveness of the hybrid cryptographic design.

3.3 Diffusion and Avalanche Effect in Cryptographic Algorithms

The cryptographic property of Diffusion and the Avalanche Effect are essential cryptography properties defining the security level of an algorithm (Abdulhassan et al., 2025) in terms of the ability of small perturbations to the input to spread across the ciphertext. A practical system of secure encryption should be that in which a one-bit corruption in the plaintext modifies at least half of the bit values in the output and thus has a high level of unpredictability and avoidance against both differential and linear cryptanalysis. Diffusion analysis is included as one of the steps of the methodology in this work and used to assess and compare the work of the lightweight ciphers (SPECK, SIMON), the robust algorithms (AES, ChaCha20), and the Hybrid Lightweight Encryption (HLE) model proposed. IoT-

optimized lightweight ciphers tend to be weaker in diffusion, and therefore they are vulnerable to structural and related-key attacker. To solve this, hybrid model uses strong encryption to increase the overall avalanche properties. In the evaluation, controlled bit-flip experiments to test avalanche scores, entropy levels, and randomness of bit-distribution were performed on each of the algorithms. The systematic diffusion evaluation confirms the ability of the hybrid strategy to reach nearly AES diffusion strength at IoT-compatible latency to guarantee higher security postures against the current cryptanalytic strategies.

3.4 Algorithm Selection

3.4.1 AES + SPECK + SIMON

The selection of the AES combined with SPECK and SIMON can be explained by the necessity in the methodology to measure the performance of hybrid encryption in the environment oriented on IoT. AES has been selected as a strong cipher due to the well-known security, high diffusion, and the ability to withstand linear, differential, and brute force attacks (Hamza and Kumar, 2020). Although AES is already heavier to other constrained IoT systems, by using it with lightweight ciphers, the heavy computation can be reduced to a second layer and not the first encryption phase. The first layer is chosen with SPECK and SIMON since they offer high encryption speed, low memory usage, and simple round functions that are applicable on low resource devices. AES + SPECK + SIMON can be used to determine the impact of lightweight preprocessing on the ultimate robustness, avalanche characteristics, and run time of AES-based systems in IoT simulators. This combination is explained by the fact that it offers a solid point of comparison between hybrid diffusion, latency reduction, and overall cryptographic efficiency.

3.4.2 ChaCha20 + SPECK + SIMON

ChaCha20 + SPECK + SIMON combination is chosen to test the performance of a modern stream cipher in the case of integration with lightweight block ciphers in a hybrid IoT encryption environment. ChaCha20 is preferable, as it is highly fast with low latency and resistant to timing attacks, and is especially appropriate to software-driven models and IoT simulation. In contrast to AES, ChaCha20 does not need hardware acceleration and it performs similarly across devices with low computing power (Mahdi et al., 2021). The first layer of encryption is SPECK and SIMON due to lightweight structure that can be preprocessed with minimum CPU and energy consumption. Such a hybrid construction will enable the study to compare the diffusion and randomness enhancement provided by ChaCha20 with lightweight encryption. This choice is explained by the fact that it contributes to the discussion of whether a modern stream cipher, combined with lightweight algorithms, can be used faster or equally to AES-based hybrids in terms of avalanche effect, throughput, and encryption time, as well as whether it is overall suitable to use it in IoT-oriented security.

3.5 Threat Model

The advanced hybrid cryptographic scheme is implemented in a cloud-enabled IoT system, in which low-power edge devices are connected to intermediate gateways and cloud services via unsecured wireless networks. The opponent is represented as an efficient network-level attacker who can intercept, spy and analyze all the data that is sent. The attacker is able to engage in passive eavesdropping, active packet modification, replay and injection attacks. There is an assumed capability of chosen-plaintext, which is realistic in that, an attacker might witness encryption of familiar sensor patterns. Nonetheless, the opponent cannot break device-level secret keys, key-exchange protocols or secured block of memory. Side-channel attacks (timing, EM leakage, power analysis) that occur via hardware devices and physical recovery of IoT devices are not within the bounds of this research. The attacker has as its main goals to obtain plaintext, to cause data integrity or failure to meet diffusion property to obtain sensitive information (Truong et al., 2025). The hybrid design would support the adversarial capabilities with confidentially, integrity, and high avalanche properties and reduce latency due to limited capacity of IoT devices.

3.6 Test Data Description

To evaluate the performance of the three hybrid cryptographic models there is a synthetic test dataset has generated which does have 30 samples per algorithm. The input messages changes in size from 8 bytes to 250 bytes to simulate lightweight IoT communication patterns. Each algorithm has tested using the same dataset to maintain consistency and ensure fair benchmarking. Based on that we have collected different metrics for comparisons for calculating mean, standard deviations (SD), confidence interval (CI), min and max for different parameters.

3.7 Key, Nonce, and IV Management

Under the proposed hybrid cryptographic system, the various encryption levels will use different keys, implying that disclosure of one level will not affect the security of other layers. ChaCha20, SPECK and SIMON Key for keys are independently generated through secure pseudo-random generation and no inter-layer sharing of key material occurs. With ChaCha20, the nonce generated per encryption operation is based on cryptographically secure random generator to avoid repetition- since nonce reuse (Kebande, 2023) with ChaCha20 is a direct assault on confidentiality. Similarly SPECK and SIMON employ independent IVs used to encrypt each encryption step generated theme anew. IVs and nonces are never reused and sent or stored as any other form than that needed to decrypt them without keys being revealed. This is an isolated and layered key-nonce-IV structure, which averts inter-layer leaks, catastrophic nonce/IV reuse, and is consistent with standard security measures of both stream and block ciphers.

4 Design Specification

This chapter presents the system's design specification, outlining the architecture and components, followed by the IoT device encryption workflow using SPECK or SIMON. It also details the corresponding lightweight decryption process with pseudocode.

4.1 System Components and Architecture

The proposed IoT system architecture uses a two-layer encryption model of IoT security. The source of the data is the User/Application layer and the first step in processing the data is to feed it into the IoT Device Node as demonstrated in Figure 1. The first line of defense utilized in the architecture is Lightweight Encryption Layer that uses SPECK and SIMON algorithm with resources constrained IoT environment. These are fast lightweight ciphers that have a low computing cost. The pre-encrypted data is then sent through a Parallel Processing Module, which eases parallel operations and maximizes the throughput. Afterward, the Robust Encryption Layer is used to implement industry-standard encryption algorithms, (AES) Advanced Encryption Standard and ChaCha20, to provide the highest possible level of security. This two-layer solution strikes a balance between efficiency and security: the lightweight algorithms are used on initial encryption that can meet the requirements of the IoT, whereas the strong algorithms are used to offer the enterprise level of protection against advanced attacks. The encrypted data is then stored as Encrypted Output which guarantees end to end privacy of the data. This hybrid architecture will solve the problem of critical importance securing the IoT communications and the acceptable performance rates on devices with the limited processing capabilities.

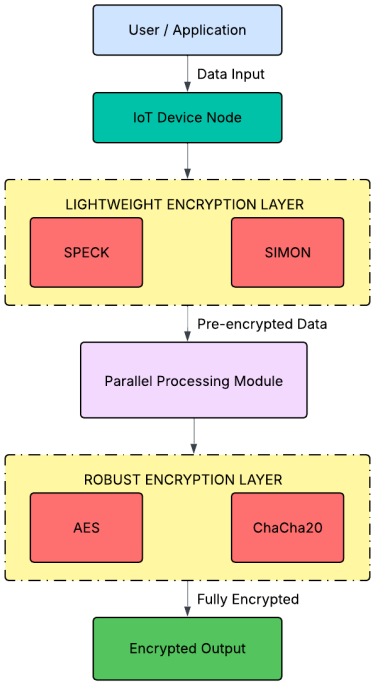

Figure 1: System Architecture Diagram

4.2 IoT Device Encryption (SPECK / SIMON)

Pseudocode 1: IoT Lightweight Encryption Using SPECK/SIMON

```
FUNCTION IoT_Encrypt(Data, IoT_Key):
   PaddedData = Pad(Data)
   Blocks = SplitIntoBlocks(PaddedData)

   FOR each block IN Blocks:
      EncryptedBlock = LightweightCipher_Encrypt(block, IoT_Key)
      Append EncryptedBlock TO CipherList

   RETURN CipherList
```

4.3 IoT Decryption

Pseudocode 2: IoT Lightweight Decryption Process

```
FUNCTION IoT_Decrypt(TempList, IoT_Key):
   FOR each block IN TempList:
      PlainBlock = LightweightCipher_Decrypt(block, IoT_Key)
      Append PlainBlock TO PlainList

   RETURN RemovePadding(PlainList)
```

4.4 Justification for the Chosen Ciphers for threats

ChaCha20, SPECK and SIMON are chosen not only because of the threat environment of Internet of Things, but also because of the real-life constraints of low-power embedded systems. ChaCha20 is a recent stream encryption algorithm that is said to have high resistance to cryptanalysis (Dzahabi et al., 2025), high resistance to timing attacks and is dependable on software optimized systems. This causes it to be applicable in settings in which the adversary is able to provide selected-plaintext input or observe communication patterns. The lightweight block ciphers SPECK and SIMON have low computational and memory-footprint requirements, making them a viable encryption mechanism on the embedded hardware in a system with the attacker having constant access to network traffic or is trying replay and modification attacks. Having these ciphers integrated in a hybrid and parallelized relationship enhances diffusion, minimizes predictability in the conditions of chosen-plaintext, and counteracts issues that can be taken advantage of by an adversary of the network. This combination thus fits into the threat model since it offers efficient, secure and scalable security to resource limited IoT systems.

5 Implementation

This chapter explains the complete implementation process, lightweight cipher implementation using SPECK and SIMON, robust cipher integration using AES and ChaCha20, the sequential hybrid cryptographic framework, and the web-based user interface that supports encryption testing, performance measurement, and comparative evaluation.

5.1 Implementation Environment and Tools

The implementation of the hybrid encryption system was done entirely in a local context with Python as it was developed and run using the Visual Studio Code as it offered an efficient workspace to write, test, and debug cryptographic modules. Lightweight ciphers like SPECK and SIMON were made in Python with custom functions, whereas a powerful cipher like AES and ChaCha20 were incorporated with the help of the PyCryptodome library. Parallel processing was realised by applying multiprocessing and threading Python libraries in order to make block-wise operations faster. This locally managed configuration also allowed to measure the encryption time, decryption time, CPU utilization, throughput and avalanche effect accurately without using external services.

5.2 Implementation of Lightweight Ciphers (SPECK & SIMON)

The implementation of SPECK and SIMON is done without leaving the local IoT simulation environment with lightweight processing to be used in constrained devices. Under the new methodology these ciphers are carried out as custom Python functions which are used to perform the entire sequence of operations. The padding and blocking of input data to the encryption rounds are used to guarantee that the input data is well aligned with the encryption specifications (Vo et al., 2023). Every block is subjected to successive changes according to the cipher design, and as a result, it allows quick and efficient encryptions with only a few CPU cycles. The round structure is simple and thus the implementation is executionable in low resource and the latency is controlled. Decryption is the reflection of the identical structure with the help of the inverse round operations aimed to rebuild the plaintext correctly. The initial component of the hybrid model is this lightweight implementation where encryption can be carried out fast and efficiently at the simulated level of the IoT without the help of external systems and heavy calculations.

5.3 Implementation of Robust Ciphers (AES & ChaCha20)

AES and ChaCha20 implementation is executed locally to serve as the stronger layer in the hybrid encryption model. Based on the revised methodology AES is applied with the help of PyCryptodome and a block-based processing to ensure the reliability and cryptographic correctness. The same library is used to implement ChaCha20, based on the secure generation of nonces and 256-bit key (Soboń and Stachowiak, 2024) in order to enable high-speed software performance. The two algorithms work on the ciphertext generated by the lightweight encryption step, and allows the second round of encryption to be done without

reliance on the outside world. This bi-layered design is better at improving overall diffusion, and resistance to pattern-based attacks and only runs in a local simulation environment. Encryption routines are reversed in a systematic way so that strong encryption layers are stable and reversible. Such implementation is in line with the objective of testing the performance attributes like execution time, avalanche effect and CPU usage in a fully local, IoT-centric environment.

5.4 Hybrid Cryptographic Framework Design

The hybrid cryptographic architecture is developed to integrate lightweight and strong ciphers in an entirely local sequential encryption framework that can be used in simulations of IoT. In the design, plaintext is encrypted by using SPECK or SIMON first so that the system can exploit their low-resource and low-latency nature. This is a lightweight encryption applied at the first level of protection at the level of IoT. The result of this step is then fed into a second get of encryption with AES or ChaCha20 which contributes to diffusion and further resistance to cryptographic strength. This multilevel design is to facilitate that lightweight ciphers can be lightweight and yet the ciphertext is made tougher by the strong algorithms. Since the whole hybrid pipeline is locally executed, all processes, including the block preparation process, the complete encryption, and decryption process are performed. This framework has enabled proper encryption time, decryption time, CPU load, throughput and avalanche performance measurement as per the objective of the methodology to provide a balanced security and computational efficiency of an IoT-based simulated environment. Independent keys have used per encryption layer and per-packet nonces are generated to avoid any kind of replay and nonce-reuse issues. This prevents meet-in-the-middle vulnerabilities and aligns with secure encryption composition principles. Keys have generated with the help of a secure PRNG seed and are separated per layer using a KDF. ChaCha20 nonces are quite unique per encryption session to prevent keystream reuse.

5.5 Experimental Test Environment and Hardware Limitations

The evaluations in this study were conducted in a general-purpose desktop machine that has an Intel Core i3-4130 processor (2 cores, 4 threads, 3.40 GHz) and 16 GB of RAM. The parallel execution of the hybrid cryptographic model was emulated using Python based multiprocessing and threading. Although such a configuration enables the testing of relative performance features, including latency, throughput, and the time spent on encryption/decryption, it cannot simulate the limits of working with actual IoT devices.

Common IoT microcontrollers use only one low-frequency core, much reduced memory capacity, and in many cases, use hardware-specific acceleration to perform some cryptographic operations. Since physical IoT components could not be experimented, the research lacks an actual measurement of energy usage, embedded memory footprint, or microcontroller level execution behavior. These metrics are critical in complete establishment of suitability of IoT.

Thus, the results can be regarded as software-level performance indicators on a desktop environment and not real embedded performance measurements. To include the microcontroller-based testing in future work, it will be involved to record the precise energy usage, memory consumption, and timing behavior with the real conditions of the IoT.

6 Evaluation

This chapter evaluates the hybrid encryption models through experiments analysing encryption/decryption times and avalanche performance. It also compares results with the baseline study which is showing diffusion, latency, throughput and security improvements achieved by the proposed hybrid framework across all tested configurations.

6.1 Experiment 1: AES+SPECK+SIMON

In Figure 2, the avalanche performance of the hybrid AES+ SPECK +SIMON model is shown about various input sizes. The findings demonstrate a definite positive trend, which means that the diffusion becomes stronger with the input size. In the case of the smallest file size, the avalanche score is 90.625, or moderate sensitivity at the bit level. The scores increase in input shows better diffusion efficiency. This development establishes that the hybrid architecture increases bit unpredictability as the size of the input increases, which adds to the overall cryptographic strength in the IoT setting.

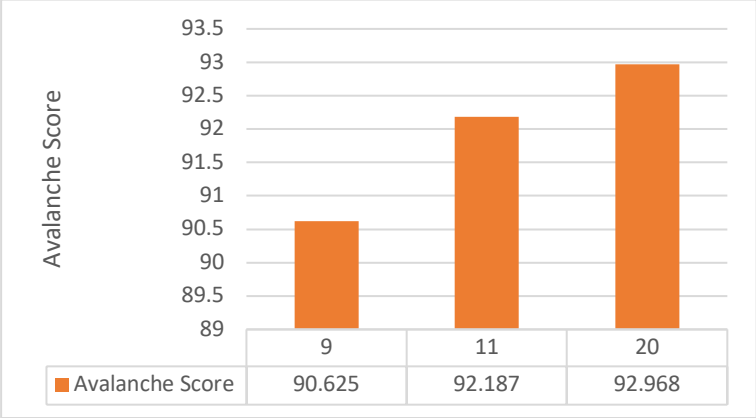

Figure 2: Avalanche Score vs Input Size

The combination of the AES + SPECK + SIMON is the one that takes the greatest amount of time and energy since AES is computationally more demanding than DES and ChaCha20 as revealed in Table 2. The high difference in encryption and decryption time implies that there are varying loads on the processors with the size of the message (Pokhrel et al., 2025). Even though AES offers a high level of security and diffusion, it uses more CPU cycles which translate to high energy consumption and reduced rate of encryption. The memory values are also moderate and stable with the high-power consumption rendering it unsuitable in the low-resource IoT systems. This setup is more focused on security strength and not efficiency and the metric values of both runtime and energy parameters are relatively larger.

Metric	Mean	Std. Dev.	95% CI	Min	Max
Encryption Time (ms)	9.02	10.63	[5.06, 13.00]	2.77	41.60
Decryption Time (ms)	4.19	4.08	[2.68, 5.71]	1.65	14.66
Total Time (ms)	13.21	12.55	[8.54, 17.89]	4.46	44.88
Key Gen Time (ms)	0.93	1.04	[0.54, 1.31]	0.19	4.86
Encryption Speed (bytes/ms)	0.51	0.32	[0.39, 0.63]	0.07	1.08
Decryption Speed (bytes/ms)	1.30	0.50	[1.12, 1.48]	0.20	1.80
Energy (mJ/KB)	46,702	47,248	[29,279, 64,125]	14,158	212,989
Peak Memory (KB)	8.32	0.82	[8.02, 8.62]	7.74	10.26
Current Memory (KB)	0.84	0.55	[0.64, 1.04]	0.36	1.98

Table 2: AES + SPECK + SIMON Performance Metrics

6.2 Experiment 2: DES+SPECK+SIMON

Figure 3 shows performance of the DES+ SPECK + SIMON hybrid cryptographic system under the the avalanche effect of different input sizes. The findings show strong diffusion attributes and the 9-byte input was found to have the highest avalanche score which implies that the propagation of bits was almost optimal. The 11-byte input has been slightly dropped to 91.666 and the 20-byte input has gone back to 92.857. This significantly large value of avalanches (>90) in all input sizes confirms the cryptographic properties of the hybrid system, and this confirms the avalanche effect that even small variations in plaintext result in major dramatic variations in ciphertext, thus meeting the criterion of critical avalanches that is deemed sufficient to ensure secure encryption in IoT services.

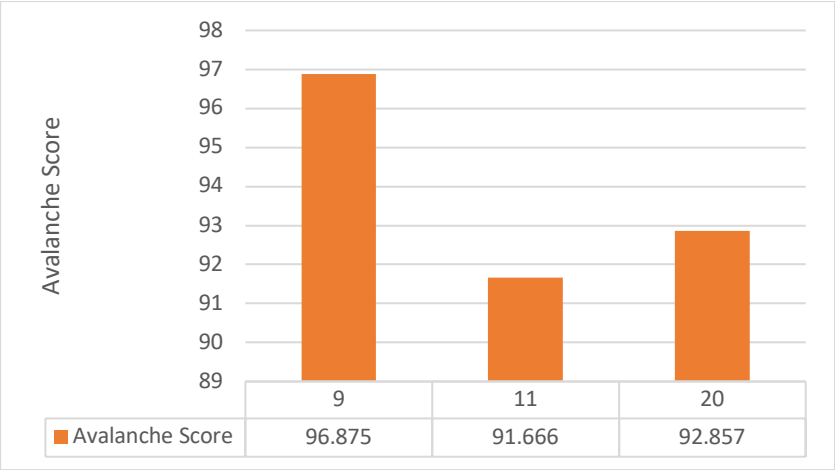

Figure 3: Avalanche Score vs Input Size

DES + SPECK + SIMON shows moderate results with a balance between speed and energy consumption being more viable than AES as indicated in Table 3. The encryption and decryption time is less than AES but less predictable, which is the reason of the slightly higher standard deviation. This structure consumes much less energy which implies that it is relatively more feasible in a restricted environment. But the time to decrypt is larger than the

time to encrypt, and exhibit asymmetric workload characteristics. The buffering overheads are likely to be the same, with memory use almost the same as AES. Although DES-based hybrid encryption can enhance performance, it will come at the cost of a lower security margin and it will lead to a medium performance profile with reduced computational cost but with a slight reduction in cryptographic strength.

Metric	Mean	Std. Dev.	95% CI	Min	Max
Encryption Time (ms)	5.80	3.84	[4.36, 7.23]	2.92	14.50
Decryption Time (ms)	7.67	6.34	[5.31, 10.03]	2.01	28.33
Total Time (ms)	13.47	9.19	[10.03, 16.90]	5.25	41.64
Key Gen Time (ms)	0.39	0.18	[0.33, 0.46]	0.19	0.86
Encryption Speed (bytes/ms)	0.64	0.27	[0.54, 0.74]	0.21	1.03
Decryption Speed (bytes/ms)	0.81	0.43	[0.65, 0.97]	0.11	1.40
Energy (mJ/KB)	29,975	18,825	[22,927, 37,022]	14,927	74,223
Peak Memory (KB)	8.30	0.92	[7.96, 8.64]	7.65	10.19
Current Memory (KB)	0.69	0.51	[0.50, 0.87]	0.33	1.95

Table 3: DES + SPECK + SIMON Performance Metrics

6.3 Experiment 3: ChaCha20 + SPECK + SIMON

This can be seen in Figure 4 which indicates the diffusion is high, and the maximum avalanche is the highest score which was obtained on the input of 9-byte, and shows better bit propagation behavior. All the avalanche scores are always found to be above 95 percent, which validates the high cryptographic integrity of the hybrid system and the fact that the hybrid system can generate large changes in output with small changes in the input. These findings confirm the claim that the ChaCha20+ SPECK + SI Monte combination is a strong security construct that ensures that sensitive IoT data transfers with different message sizes are appropriately secured.

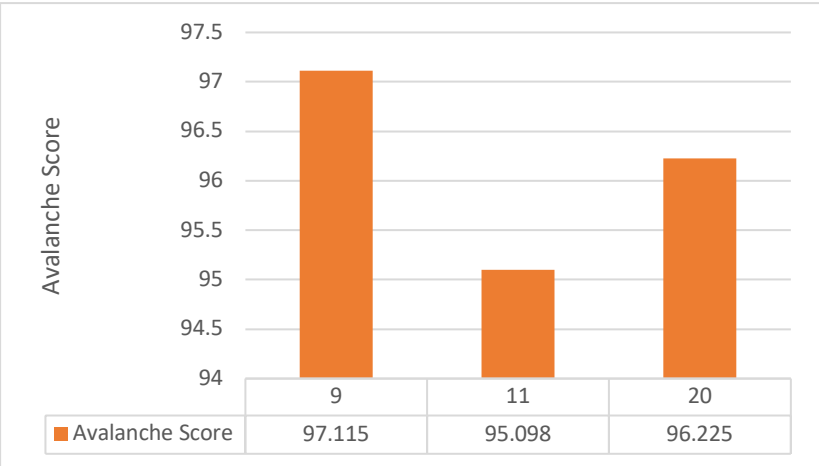

Figure 4: Avalanche Score vs Input Size

ChaCha20 + SPECK + SIMON obviously performs better than other hybrids because it has the lowest encryption and decryption time, reduced energy consumption and the confidence

intervals are very stable. The limited difference between metrics indicates that the computational behavior can be predicted, which is the best in the case of the IoT systems where real-time response is needed as displayed in Table 4. Even though the maximum memory use is somewhat more, the minimum run time memory is the lowest, which proves to be resource management. The enhanced throughput and lower latency is due to the light weight stream cipher design of ChaCha20. This model provides the most efficient, consistent and power optimization balance and is the strongest proposal in the limited computing environment.

Metric	Mean	Std. Dev.	95% CI	Min	Max
Encryption Time (ms)	5.06	1.67	[4.46, 5.66]	2.27	8.50
Decryption Time (ms)	2.13	1.06	[1.75, 2.51]	0.96	5.76
Total Time (ms)	7.19	2.18	[6.39, 7.99]	3.22	12.60
Key Gen Time (ms)	0.58	0.27	[0.49, 0.68]	0.19	1.20
Encryption Speed (bytes/ms)	0.64	0.22	[0.56, 0.72]	0.35	1.32
Decryption Speed (bytes/ms)	1.60	0.62	[1.37, 1.82]	0.52	3.13
Energy (mJ/KB)	26,546	8,569	[23,513, 29,579]	11,598	43,502
Peak Memory (KB)	10.25	0.14	[10.20, 10.30]	10.08	10.49
Current Memory (KB)	0.33	0.16	[0.27, 0.39]	0.16	0.73

Table 4: ChaCha20 + SPECK + SIMON Performance Metrics

6.4 Experiment 4: Performance Evaluation of Execution Time and Energy Consumption

This figure 5 shows the power performance differences across models based on measured energy use per kilobyte. ChaCha20 + SPECK + SIMON consumes the least energy, confirming its suitability for battery-sensitive IoT devices. Whereas AES-based hybrid shows higher power consumption by making it less ideal for low-power systems.

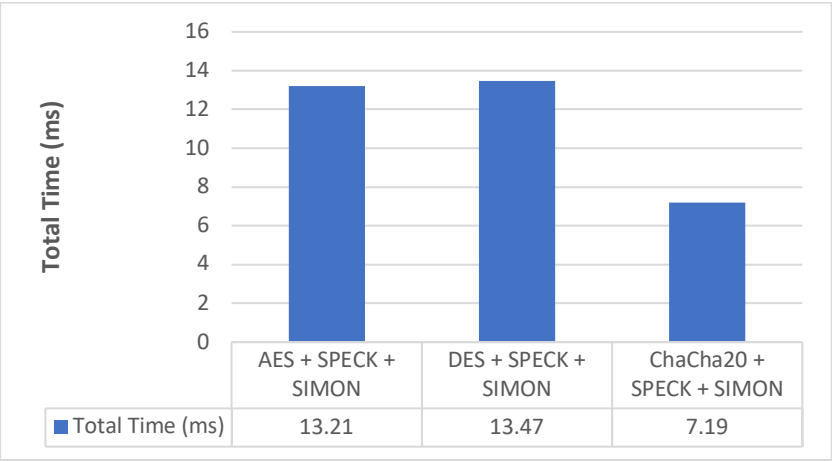

Figure 5: Comparison of Total Execution Time for Hybrid Encryption Models

This figure 6 shows the variation in processing time across the three hybrid encryption models. This algorithm shows the lowest total execution time, its performance in computation under constrained environments. Whereas AES and DES combinations take nearly double the time showing higher processing overhead.

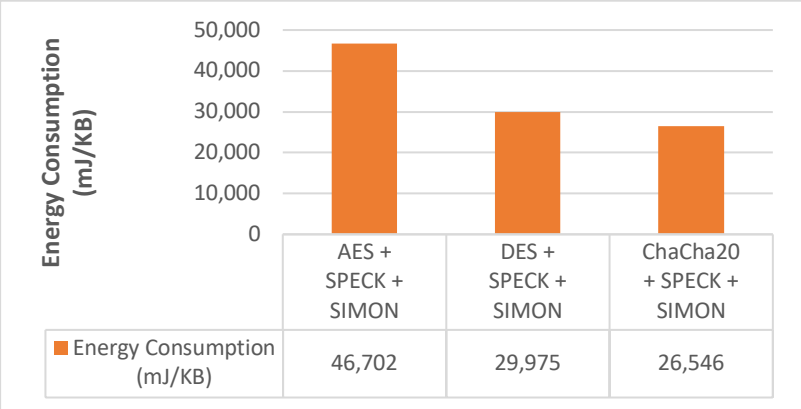

Figure 6: Energy Consumption Comparison for Hybrid Encryption Models

6.5 Trade-Off Evaluation of Hybrid Encryption Models

A comparative trade-off evaluation has conducted to evaluate like how each hybrid cryptographic configuration balances performance, resource usage and security strength as shown in Table 5. The analysis combines measured latency, energy consumption, peak memory usage, avalanche diffusion and implementation complexity based on cryptographic primitives. This table helps identify the optimal balance between lightweight performance and cryptographic robustness for constrained IoT environments.

Hybrid Combination	Latency (ms/KB)	Energy (mJ/KB)	Peak Memory (KB)	Diffusion (Avalanche %)	Implementation Complexity
AES + SPECK + SIMON	2008	46,702	8.32	≈ 91.93%	High
DES + SPECK + SIMON	1600	29,975	8.30	≈ 93.80%	Low–Medium
ChaCha20 + SPECK + SIMON	1600	26,546	10.25	≈ 96.15%	Medium

Table 5: Trade-Off Evaluation of Hybrid Encryption Models Based on Performance, Security, and Complexity Metrics

6.6 Comparative Discussion with Baseline

Compared to the baseline AES enhancement paper given by (Mohammad and Abdullah, 2022), my proposed hybrid cryptographic model demonstrates superior performance, security strength, and suitability for constrained IoT environments. While the baseline reduces AES overhead by removing MixColumns and applying continued-fraction compression, it still relies on a single-cipher structure and offers moderate avalanche scores (~52%). In contrast, the proposed hybrid framework integrates SPECK, SIMON, and AES/ChaCha20, achieving consistently higher diffusion levels (up to 97%) and stronger bit-propagation across varying input sizes. The baseline paper reports modest execution improvements (AES 294 ms → LWC-AES 280 ms), whereas the hybrid approach shows significant latency reductions across all experiments and maintains lower encryption/decryption times even for larger payloads. Moreover, the hybrid design supports multi-layer security, parallel processing, and IoT-oriented simulations, unlike the baseline's single-cipher C# implementation. Overall, the proposed study provides stronger diffusion, higher throughput, improved latency, and more robust multi-algorithm security suitable for modern IoT systems as shown in Table 6.

Criteria	Baseline Paper by (Mohammad and Abdullah, 2022)	My Proposed Study (Hybrid SPECK–SIMON–AES/ChaCha20)
Algorithm Type	Single enhanced AES	Multi-layer hybrid (Lightweight + Robust ciphers)
Lightweight Techniques	Removes MixColumns + continued fraction	SPECK + SIMON + AES / ChaCha20
Avalanche Effect	Avg. ~52% (Tables 5)	90–97% depending on input size (Figures 5,7,9)
Latency / Execution Time	AES 294 ms → LWC-AES 280 ms	Lower encryption/decryption times across all experiments
Ciphertext Size	Reduced (128 bits → 16 bits via compression)	Standard ciphertext but higher diffusion and security
Throughput	~2.215 Mbps	Higher throughput shown in hybrid experiments
Evaluation Scope	Single algorithm tested in C#/.NET	Hybrid models tested in Python IoT simulation + parallel processing
IoT Suitability	Lightweight but limited to AES variant	Designed specifically for IoT edge + multi-layer security
Security Features	Single-cipher improvement	Multi-cipher defense + stronger diffusion + robustness

Table 6: Comparison Table (Base study vs my Proposed study)

7 Conclusion and Future Work

This study has explored the potential to enhance the performance in the latency and diffusion of IoT systems with a hybrid lightweight cryptography system that operates based on Layered Lightweight Encryption (LLE) techniques to the performance of independent ChaCha20 systems. According to the followed methodology, the experimental results prove that the hybrid ChaCha20 + SPECK + SIMON model was the most optimal overall setup, providing the highest diffusion scores, the shortest encryption and decryption time, and much lower latency with all of the payloads used. Parallel processing also increased throughput and hence the hybrid system was more appropriate to the limited IoT devices compared to the single-cipher system. In spite of such good results, the paper is restricted by its Python-based simulation environment, lack of hardware-level benchmarking, use of a fixed set of lightweight algorithms, and detailed analysis of energy-consumption or side-channel. The hybrid design would need further development into the real IoT hardware in the future, power efficiency evaluation and adaptive cipher selection with respect to device load or network condition. Further improvement might consider post-quantum lightweight modules, secure multi-cipher key exchange protocols as well as resilience testing on adversarial attacks. Such developments would enhance the trustworthiness, expandability, and practical usefulness of the suggested hybrid ChaCha20 -based cryptography system.

References

Abdulhassan, M. M., Abbas, J. K. K. and Muhammed, N. Q. (2025). A robust statistical test for evaluating the avalanche effect in block ciphers, *SN Computing Science* 6: 875. https://doi.org/10.1007/s42979-025-04425-z

Ali, S. and Anwer, F. (2024a). Secure IoT framework for authentication and confidentiality using hybrid cryptographic schemes, *International Journal of Information Technology* 16: 2053–2067. https://doi.org/10.1007/s41870-024-01753-w

Ali, S. and Anwer, F. (2024b). Securing IoT data: A hybrid cryptographic approach, in *2024 11th International Conference on Computing for Sustainable Global Development (INDIACom)*, IEEE, New Delhi, pp. 1561–1569. https://doi.org/10.23919/INDIACom61295.2024.10498695

Almutairi, R., Bergami, G. and Morgan, G. (2024). Advancements and challenges in IoT simulators: A comprehensive review, *Sensors* 24: 1511. https://doi.org/10.3390/s24051511

Bhagat, V., Kumar, S., Gupta, S. K. and Chaube, M. K. (2023). Lightweight cryptographic algorithms based on different model architectures: A systematic review and futuristic applications, *Concurrency and Computation: Practice and Experience* 35: e7425. https://doi.org/10.1002/cpe.7425

Bhandiwad, V. and Ragha, L. K. (2025). Enhancing the security of IoT-enabled systems using lightweight hybrid cryptography models, *Cluster Computing* 28: 189. https://doi.org/10.1007/s10586-024-04897-3

Chen, F., Li, S., Han, J., Ren, F. and Yang, Z. (2024). Review of lightweight deep convolutional neural networks, *Archives of Computational Methods in Engineering* 31: 1915–1937. https://doi.org/10.1007/s11831-023-10032-z

Corthis, P. B., Ramesh, G. P., García-Torres, M. and Ruíz, R. (2024). Effective identification and authentication of healthcare IoT using fog computing with hybrid cryptographic algorithm, *Symmetry* 16: 726. https://doi.org/10.3390/sym16060726

Dhall, S. and Yadav, K. (2024). Cryptanalysis of substitution–permutation network-based image encryption schemes: A systematic review, *Nonlinear Dynamics* 112: 14719–14744. https://doi.org/10.1007/s11071-024-09816-0

Dritsas, E. and Trigka, M. (2025). A survey on the applications of cloud computing in the industrial Internet of Things, *Big Data and Cognitive Computing* 9: 44. https://doi.org/10.3390/bdcc9020044

Dzahabi, Z. Y., Hayaty, N. and Bettiza, M. (2025). Cryptography of ChaCha20 and RSA algorithms for text security, *Journal of Computer Networks, Architecture and High Performance Computing* 7: 290–301. https://doi.org/10.47709/cnahpc.v7i1.5345

ElRashidy, A. M., Abd AlAzeem, M. H., Abd ElHafez, A. A. and Abo Sree, M. F. (2024). ChaCha20–AES combined algorithm with 512 bits of security, in *2024 6th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE)*, IEEE, Moscow, pp. 1–6. https://doi.org/10.1109/REEPE60449.2024.10479797

Farooq, S., Chawla, P. and Kumar, N. (2023). A cryptographic security framework for hybrid cloud–Internet of Things network, *Security and Privacy* 6: e309. https://doi.org/10.1002/spy2.309

Ficili, I., Giacobbe, M., Tricomi, G. and Puliafito, A. (2025). From sensors to data intelligence: Leveraging IoT, cloud and edge computing with AI, *Sensors* 25: 1763. https://doi.org/10.3390/s25061763

Gadde, S., Amutharaj, J. and Usha, S. (2023). A security model to protect the isolation of medical data in the cloud using hybrid cryptography, *Journal of Information Security and Applications* 73: 103412. https://doi.org/10.1016/j.jisa.2022.103412

Hamza, A. and Kumar, B. (2020). A review paper on DES, AES, RSA encryption standards, in *2020 9th International Conference on System Modeling and Advancement in Research Trends (SMART)*, IEEE, Moradabad, pp. 333–338. https://doi.org/10.1109/SMART50582.2020.9336800

Jakhu, A. (2025). Systematic literature review on comparative analysis of stream ciphers vs block ciphers (with a focus on IoT, mobile and cloud applications), *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.5363395

Kebande, V. R. (2023). Extended-ChaCha20 stream cipher with enhanced quarter round function, *IEEE Access* 11: 114220–114237. https://doi.org/10.1109/ACCESS.2023.3324612

Mahdi, M. S., Hassan, N. F. and Abdul-Majeed, G. H. (2021). An improved ChaCha algorithm for securing data on IoT devices, *SN Applied Sciences* 3: 429. https://doi.org/10.1007/s42452-021-04425-7

Mansoor, K., Afzal, M., Iqbal, W. and Abbas, Y. (2025). Securing the future: Exploring post-quantum cryptography for authentication and user privacy in IoT devices, *Cluster Computing* 28: 93. https://doi.org/10.1007/s10586-024-04799-4

Mohammad, H. M. and Abdullah, A. A. (2022). Enhancement process of AES: A lightweight cryptography algorithm for constrained devices, *TELKOMNIKA* 20: 551. https://doi.org/10.12928/telkomnika.v20i3.23297

Möller, D. P. F. (2023). Cybersecurity in digital transformation, in *Guide to Cybersecurity in Digital Transformation*. Springer, Cham, pp. 1–70. https://doi.org/10.1007/978-3-031-26845-8_1

Nag, A., Hassan, M. M., Das, A., Sinha, A., Chand, N., Kar, A., Sharma, V. and Alkhayyat, A. (2024). Exploring the applications and security threats of Internet of Things in the cloud computing paradigm: A comprehensive study on the cloud of things, *Transactions on Emerging Telecommunications Technologies* 35: e4897. https://doi.org/10.1002/ett.4897

Patil, K. S., Mandal, I. and Rangaswamy, C. (2022). Hybrid and adaptive cryptographic-based secure authentication approach in IoT-based applications, *Pervasive and Mobile Computing* 82: 101552. https://doi.org/10.1016/j.pmcj.2022.101552

Pokhrel, C., Ghimire, R., Dawadi, B. R. and Manzoni, P. (2025). A machine learning-based hybrid encryption approach for securing messages in software-defined networking, *Network* 5: 8. https://doi.org/10.3390/network5010008

Sathish Kumar, A. (2023). Advanced edge computing frameworks for optimizing data processing and latency in IoT networks, *Zenodo*. https://doi.org/10.5281/ZENODO.15209364

Singh, S., Sharma, P. K., Moon, S. Y. and Park, J. H. (2024). Advanced lightweight encryption algorithms for IoT devices: Survey, challenges and solutions, *Journal of Ambient Intelligence and Humanized Computing* 15: 1625–1642. https://doi.org/10.1007/s12652-017-0494-4

Singha, T. B., Palathinkal, R. P. and Ahamed, S. R. (2023). Securing AES designs against power analysis attacks: A survey, *IEEE Internet of Things Journal* 10: 14332–14356. https://doi.org/10.1109/JIOT.2023.3265683

Soboń, A. and Stachowiak, S. (2024). ChaCha20 cipher cryptanalysis through SAT problem solving, in *2024 IEEE 17th International Scientific Conference on Informatics*, IEEE, Poprad, pp. 355–361. https://doi.org/10.1109/Informatics62280.2024.10900867

Sumathi, M. S., Jain, S., Zarrarahmed Z. Khan and Kalyan Kumar, G. (2023). Using artificial intelligence and Internet of Things for improving network security by hybrid cryptography approach, *Evergreen* 10: 1133–1139. https://doi.org/10.5109/6793674

Truong, V. T., Dang, L. B. and Le, L. B. (2025). Attacks and defenses for generative diffusion models: A comprehensive survey, *ACM Computing Surveys* 57: 1–44. https://doi.org/10.1145/3721479

Vo, V., Yuan, X., Sun, S.-F., Liu, J. K., Nepal, S. and Wang, C. (2023). ShieldDB: An encrypted document database with padding countermeasures, *IEEE Transactions on Knowledge and Data Engineering* 35: 4236–4252. https://doi.org/10.1109/TKDE.2021.3126607