National College of Ireland

ASSET: A Parallel Lightweight Cryptographic Framework for IoT and Cloud Security

MSc Research Project
Cloud Computing

Amrutha Shivashankaramurthy
Student ID: 23172151

School of Computing
National College of Ireland

Supervisor: Prof. Diego Lugones

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Amrutha Shivashankaramurthy
Student ID:	23172151
Programme:	Cloud Computing
Year:	2025
Module:	MSc Research Project
Supervisor:	Prof. Diego Lugones
Submission Due Date:	26/05/2025
Project Title:	ASSET: A Parallel Lightweight Cryptographic Framework for IoT and Cloud Security
Word Count:	7188
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:	Amrutha Shivashankaramurthy
Date:	26th May 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):	

ASSET: A Parallel Lightweight Cryptographic Framework for IoT and Cloud Security

Amrutha Shivashankaramurthy
23172151

Abstract

Lightweight cryptography is very important for securing Internet of Things (IoT) environments which demands encryption models that balance performance, diffusion and latency. Traditional models like Advanced Encryption Standard (AES) while highly secure, suffer from high computational costs, making them less suitable for real-time IoT applications. Lightweight alternatives such as SPECK and SIMON gives better performance but compromise on security and avalanche effect. Traditional models proposed a modified AES-based lightweight cryptographic scheme (LWC-AES) that partially addressed performance issues by removing MixColumns and using continued fractions. However, their approach remained sequential, resulting in persistent latency and reduced diffusion. This study introduces a novel parallel hybrid cryptographic model that combines AES, SPECK, and SIMON (ASSET) in a multi-threaded architecture. Unlike prior layered methods, this model encrypts plaintext in parallel segments, drastically reducing latency and enhancing diffusion through diversified cipher strengths—AES for security, SPECK for speed, and SIMON for diffusion. It has been implemented using Python, Flask and AWS Cloud9. The model performs good than LWC-AES in encryption/decryption speed, avalanche effect and CPU usage. This study gives a scalable and secure cryptographic solution for high-performance, real-time cloud and IoT environments.

Keywords: **Lightweight Cryptography, Diffusion, Cloud, Internet of Things (IoT), SPECK, SIMON**

1 Introduction

The rapid expansion of the Internet of Things (IoT) has created a growing demand for encryption methods that are not only secure but also lightweight and powerful Singh et al. (2024). While traditional standards like AES provide strong security, they are often too resource-intensive for low-power IoT devices. In contrast, lightweight ciphers such as SPECK and SIMON offer faster performance and a smaller hardware footprint, though with slightly lower diffusion capabilities. Most current encryption solutions rely on sequential, layered models that increase processing time due to their step-by-step nature. This research introduces a novel parallel hybrid encryption model that integrates ASSET which is been designed to improve encryption speed, enhance diffusion (avalanche effect), and maintain robust security, making it ideal for real-time IoT environments.

1.1 Background

Conventional lightweight cryptographic models either compromise on diffusion (security) or impose latency due to sequential processing. A hybrid cryptographic scheme such as ASSET can leverage the strengths of each cipher, but current layered implementations suffer from high processing time. Thus, there is a need to design a parallel cryptographic system that maintains high security, minimizes latency, and improves diffusion for real-time IoT environments.

1.2 Research Questions

Can a parallel hybrid Lightweight encryption model combining AES, SPECK, and SIMON reduce encryption latency while enhancing diffusion (avalanche effect) and maintaining low computational complexity for cloud environments?

It is rooted in the practical challenges which is been faced by low- powered IoT systems and edge computing environments. In real-world scenarios especially in embedded systems and sensor-based devices whose processing power, memory capacity and energy consumption are very limited. While it is theoretically possible to achieve near-zero latency by using ultra-lightweight or weakened encryption methods by doing so compromises the security and diffusion strength which are very important for data integrity and resistance to cryptanalysis. This study recognizes that not all devices require ultra-low latency at the cost of security. Many IoT and edge devices can tolerate moderate and predictable latency provided they receive a secure and lightweight encryption mechanism that does not overwhelm their limited resources. Hence the proposed parallel hybrid model (ASSET) strategically combines AES (for strong encryption), SPECK (for speed), and SIMON (for enhanced diffusion) which is executed in a parallelized structure that optimizes for both speed and cryptographic strength. This approach targets a middle ground—neither sacrificing encryption strength nor overburdening low-resource hardware. It is designed for scenarios where latency must be minimized but not eliminated at the expense of security, ensuring the model is scalable, secure, and efficient for real-time data transmission in cloud-connected IoT ecosystems. Thus, the research question is not just about improving latency—it is about balancing performance, security, and device constraints to suit the demands of modern lightweight cryptography applications.

1.3 Objectives of the Research

The objectives of the research are:

- To develop a lightweight parallel hybrid cryptographic model using ASSET optimized for real-time data protection in IoT and cloud environments.
- To implement a parallel processing approach using multi-threading to execute ASSET encryption simultaneously, aiming to minimize latency and maximize performance compared to traditional sequential models.
- To evaluate the proposed lightweight parallel encryption system in terms of encryption/decryption time, avalanche effect, and computational performance and benchmark it against standalone and sequential lightweight cryptographic techniques

1.4 Structure of the Report

This report is structured into seven chapters to clearly explain the problem and the developed solution.

- Introduction: Provides the background, defines the problem in current lightweight cryptographic methods, and states the research objectives and questions driving the study.
- Literature Review: Reviews the evolution of cryptography, especially focusing on AES, SPECK, and SIMON in the context of IoT, identifying research gaps in performance and diffusion.
- Research Methodology: Discusses the selection and justification of the cryptographic algorithms used (AES, SPECK, SIMON), detailing the proposed parallel encryption strategy.
- Design Specification: Describes the system design including tools, architecture, and performance measurement techniques such as latency, diffusion, and resource utilization.
- Implementation: Explains how the proposed hybrid cryptographic model was developed using Python and Flask, detailing encryption and decryption processes and integration.
- Evaluation: Presents the performance outcomes of ASSET versus traditional models using case studies, analysing throughput, avalanche effect, and resource consumption.
- Conclusion and Future Works: Summarizes the contributions and advantages of the ASSET model, with suggestions for future enhancements and real-world deployments in IoT hardware

2 Related Work

2.1 Evolution of Cryptographic Techniques

Cryptographic techniques have developed through an ongoing transformation because modern society needs secure communication against increasing technological security threats Chahar (2025). The earliest cryptographic practice developed in ancient civilizations through basic substitution ciphers that used the Caesar Cipher method by shifting letters for message protection. The basic secret encryption provided by classical methods became vulnerable when attackers performed frequency analysis attacks. Scientific cryptography took shape in the twentieth century through wartime efforts that produced the Enigma machine and its resulting impact on code-breaking and the ultimate creation of computer science. Cryptography evolved into a mathematical process after World War II resulting in symmetric key cryptography algorithms such as DES (Data Encryption Standard) being developed in the 1970s stated by Naser (2021). Symmetric cryptography faced a crucial distribution key challenge that led to the invention of public-key cryptography in the late 1970s through the development of the RSA algorithm. Virtual communication security became possible without previous key sharing

because of this advancement which also established digital signature protocols that advanced secure e-commerce environments. During the 1990s researchers developed AES (Advanced Encryption Standard) which succeeded DES while providing stronger encryption and better efficiency capabilities. Hash functions alongside digital signatures became essential elements for data authentication and integrity checks in public and private systems at the same time. Internet and mobile technology advancements demanded the creation of SSL/TLS cryptographic protocols because they were needed to secure online communication.

2.2 Overview of Cryptographic Techniques in IoT

2.2.1 Advanced Encryption Standard (AES)

In the context of the Internet of Things (IoT) secure communication is a very important challenge of devices and the growing volume of sensitive data exchange. Traditional cryptographic algorithms like AES are mostly considered too heavy for such environment to explore lightweight adaptations In the first study which has been given by Sultan et al. (2020) provides a lightweight cryptographic approach based on AES variants to strengthen security protocols for IoT environments focused on wireless sensors. This study focuses on resolving the difficulty of traditional type of cryptographic techniques because they fail to work good for IoT systems due to their high requirements for processing power and memory as well as energy consumption. The authors used AES-128, AES-192 and AES-256 variants within an experimental network constructed for data transmission and packet delivery without compromising reliability. Power analysis served as the method to determine the energy performance of these different variants. The combination of AES-192 and AES-256 with 8 rounds achieves a security level that consumes low power thus proving suitable for small IoT devices in a good way.

The study proposed by Salman et al. (2022) compares different lightweight AES-128 block cipher modifications by looking appropriate encryption solutions for IoT devices with limited computational resources. The study has been investigated AES-128 through the evaluation of diverse modifications implemented from 2016 to 2021 based on security performance alongside encryption speed and interface length scalability. The proposed cryptographic solution modifies ShiftRow operation and merges MixColumn operation with AddRoundKey operation into one cycle along with decreasing the number of encryption rounds to six. The researchers evaluated these modifications to determine their performance for performance optimization as well as security enhancement in lightweight cryptographic platforms. The optimized variants in the results research show superior performance compared to other assessed techniques regarding execution speed and system protection standards. The research study faces a major limitation because it concentrates on AES-128 encryption and particular modifications while excluding broader lightweight cryptographic requirements present in complete IoT application frameworks.

The researcher Arpaia et al. (2020) plans an experimental analysis of AES-128 encryption in IoT sensor networks when confronted by severe signal-to-noise ratio conditions. The proposed study uses an improved form of scatter attack for side-channel analysis on AES-128 encryption to show its potential vulnerability to attacks originating from sensor power consumption measurements. There is an approach which integrates a commercial data acquisition system along with statistical analysis through basic digital preprocessing to obtain encryption keys from an 8-bit IoT microcontroller. The experimental research has proven that attackers operating micro-transducers gain access to AES-128 keys which

demonstrates a key flaw in IoT encryption standards today. A standardized experimental method becomes vital for obtaining complete IoT security assessments because the findings demand this standardization. This method faces a key restriction since it requires particular hardware configurations and attack scenarios that need enhanced countermeasures for optimal adaptation across various IoT operating environments.

A secure communication framework for IoT systems was implemented practically by Al-Mashhadani and Shujaa (2022) through the use of AES to protect IoT environments against their resource constraints and complexity. The research establishes an operational implementation of encryption and decryption procedures for sensor information transmission during real-time systems. The system connects IoT sensors with an ESP32 microcontroller which engages in data encryption before sending it either to the user-designated internet address or to a publicly accessible internal ESP32 network IP. The receiving end implements a decryption procedure to obtain original sensor values. The method maintains a protective balance alongside operational speed which proves that AES works well to secure limited-power IoT system implementations. A limitation emerges from this research because it utilizes exclusive hardware module ESP32 together with restricted testing of scalability which impacts the study's potential reach for multiple and diverse IoT networks.

The approach developed by Ahmad et al. (2024) creates a specific AES-128 cryptographic model for IoT devices which uses 90nm CMOS technology to protect scarce resources while ensuring security in IoT applications. The study's purpose focuses on improving AES encryption performance and minimizing area requirements to enable real-time IoT data communications without sacrificing data security. The Verilog HDL development includes a Synopsys tool validation of an updated S-Box structure which delivers enhanced security while minimizing space utilization and accelerating information transmission speed. Simulation and synthesis of the design result in a throughput rate of 14.54 Mbps with a 100 MHz clock frequency and the implementation requires 0.4324 mm^2 of space to function correctly within IoT hardware constraints. The designed purpose-built solution provides an ideal balance regarding performance and hardware utility consumption. Despite its valuable results the study is restricted because it concentrates on the 90nm fabrication technology and provides limited details about power utilization which remains critical for the operation of battery-powered Internet of Things devices.

At last Mohammad and Abdullah (2022) proposed an enhanced version of the Advanced Encryption Standard (AES) tailored for restricted devices characterized by limited memory, simple processors, and constrained power supply. The primary aim of the study was to develop a lightweight cryptographic algorithm that ensures data confidentiality and integrity while minimizing computational overhead, making it suitable for resource-constrained environments. To achieve this, the researchers introduced a Lightweight Cryptography-AES (LWC-AES) algorithm that eliminates the mix column transformation—thereby reducing complexity—and applies a mathematical function based on continued fractions to compress and obfuscate the ciphertext, improving both encryption speed and data transmission efficiency. The proposed method was tested using performance metrics such as encryption execution time and the avalanche effect, showing that LWC-AES outperformed traditional AES by reducing execution time (e.g., encrypting 45.1 KB of data in 280 ms versus 294 ms with AES) while maintaining acceptable levels of security. However, the approach may be limited by the lack of in-depth comparison with other state-of-the-art lightweight cryptographic schemes, and it does not detail the computational trade-offs involved in the continued fraction transformation, which could

impact scalability and interoperability across different device architectures. These research gaps provides foundation for the current study.

2.2.2 SPECK Cipher

Lightweight cryptography (LWC) has emerged as a critical field in addressing the security challenges faced by resource-constrained Internet of Things (IoT) environments, where traditional cryptographic algorithms fall short due to their high computational and memory demands. Various studies have explored the application and enhancement of lightweight encryption techniques, particularly the Speck algorithm introduced by the NSA.

First study given by Sleem and Couturier (2021) who describe Speck-R which is as an ultra-lightweight cryptographic algorithm made for Internet of Things environments to overcome common cryptographic scheme issues with power consumption and memory usage and processing time. The NSA-developed Speck cipher received an upgrade in Speck-R which combines ARX-based structures with an additional key-dynamic substitution module that uses dynamic keys. Speck-R introduces security-preserving functionality by reducing round counts from 26 to 7 rounds which significantly enhances processing speed. The research tests its performance using multiple statistical and cryptographic tests while implementing hardware on three popular IoT devices to validate findings. The system execution time decreases by between 18 and 77 percentage according to the evaluation while maintaining better performance than original Speck implementation. A weakness of Speck-R exists in its key management system that depends on the dynamic substitution mechanism because it results in implementation complexity issues across different platforms. This research establishes Speck-R as a reliable and high-performance encryption standard that works effectively for limited IoT platforms.

The authors in Dhakare et al. (2024) evaluate the use of lightweight cryptographic algorithms for protecting IoT devices because traditional methods with their high computational requirements do not work in resource-limited IoT systems. The study delivers extensive research about lightweight cryptographic techniques through an implementation and evaluation of Advanced Encryption Standard (AES) and Speck algorithms for IoT devices. The research display how these lightweight ciphers boost IoT security by investigating their implementation in both embedded hardware and software while maintaining device operational performance. The research findings show that AES and Speck require less memory storage than typical algorithms which supports their practical use for IoT system security. The study presents strong evidence on performance benefits in limited spaces yet it fails to perform thorough security evaluations and complete tests against multiple possible attacks which would better demonstrate lasting device fortification measures. The research proves that AES and Speck serve as operative cryptographic mechanisms for protecting IoT devices.

The authors in AbdulRaheem et al. (2022) developed an efficient lightweight encryption technique implementing Speck for edge-IoT-based smart healthcare systems (SHS) to secure the low-latency processing of smart sensor-generated medical data. Edge computing promotes faster communication and computation which helps overcome the processing and real-time transmission challenges created by substantial health data amounts. The implementation method merges Speck encryption with IoT-edge systems to create faster encryption times by achieving security objectives without sacrificing performance efficiency. The results of the study show that Speck-based encryption demonstrates superior

performance compared to other methods when applied to medical data in edge-enabled systems based on accuracy and memory and time metrics and precision evaluation. Health information security reaches its peak through the method which delivers increased confidentiality and integrity together with availability for time-sensitive medical choices. A main weakness in this work exists in the limited evaluation approach that lacks thorough assessments of cryptographic strength alongside missing deployments in actual clinical settings. The study demonstrates how Speck works as an appropriate lightweight encryption model for upcoming smart healthcare systems.

Another study proposed by Altaie and Hoomod (2023) present a security solution that uses SPECK and PRESENT encryption standards to protect IoT monitoring systems in enterprise construction and critical infrastructure deployments. The research intends to build a multi-layered defensive system protecting sensor data security while safeguarding it from theft incidents and fire damage and data breaches in educational organizations along with facilities in factories and healthcare centers. The system performs sensor data encryption as its initial step before SPECK re-encryption occurs on Raspberry Pi devices that subsequently use MQTT protocol to transmit secure data to a central PC. Among its features the system employs data classification technology to sort information between urgent processing then fast processing before regular data for prompt processing of essential information. This system demonstrates safe encryption methods that protect privacy and consume minimal resources thus being suited for edge systems and constrained devices which protects data while keeping it complete and secure. The research lacks essential evaluation data about performance amounts and power consumption as well as current cryptographic attack analysis methods. Authentication between devices becomes more secure because the dual-encryption scheme delivers extra safeguards to smart IoT environments.

2.2.3 SIMON Cipher

In recent years the increasing adoption of lightweight encryption algorithms has become as a good solution for securing data in Internet of Things (IoT) and cloud-based environments. Among these the SIMON cipher has been originally introduced by the NSA. There are so many studies which have explored the cryptographic strength and practical implementation of SIMON. Dwivedi and Srivastava (2023) conduct a study on the security perspective of Feistel-based block ciphers named SIMON developed by U.S. NSA (2013) and SIMECK created by the University of Waterloo which usually intended for IoT encryption systems. The research evaluates differential cryptanalysis resistance of these ciphers by analyzing possible differential characteristics but evaluation difficulties exist due to lengthy computational requirements. A heuristic approach combining nested tree searches serves the authors to successfully discover differential paths. The proposed method enhances cryptanalysis by increasing speed and decreasing the number of rounds yet producing optimal encryption results. The proposed method delivers effective differential paths for both ciphers through a reduced framework compared to conventional exhaustive techniques. It achieves these results while using a dependable algorithm. The key limitation of this analysis involves examining reduced-round versions of the ciphers since such versions might not accurately represent their full-round security aspects thus restricting its real-world applicability.

The authors designed a security framework which utilized lightweight cryptographic methods to augment IoT data privacy in mobile commercial environments by Taware

et al. (2021). Two complementary strategies were developed to protect mobile-commerce data transmission: an FFBN algorithm separated confidential information for enhanced identification and subsequent SIMON block cipher encryption of those data. Researchers implemented an optimization process through meta-heuristic Crow Search Algorithm (CSA) key generation which led to development of the SIMON-CSA model. The researchers managed to better protect data security with a technique that shortened encryption key production periods and preserved its highly secretive nature. The results of this study remain under question because the publication was invalidated and because there was no practical testing of the methodology despite technical issues possibly affecting reproducibility and methodology quality and originality.

The SIMON lightweight encryption algorithm receives recommendation from Khalifa et al. (2024) for protecting IoT sensor nodes transmission data in IoT environments because of growing security challenges from increasing device connectivity. The research project designs and simulates the performance of the SIMON encryption algorithm to establish its effectiveness for restricted IoT environments. The authors measure encryption performance with process run times and CPU activity to demonstrate that SIMON creates an appropriate cryptographic solution for minimal power handheld IoT devices. One of the main limitations involves the need for improved efficiency enhancements because SIMON delivers promising performance yet requires additional performance optimization when used under complex real-world operational conditions. The paper also lacks analysis on the algorithm's ability to defend against targeted attacks for wider acceptance to occur.

Another study given by Muthumari et al. (2021) establish a high-security framework for deduplicated big data storage in cloud infrastructure by applying dual encryption through Optimal SIMON Cipher (OSC) as its core mechanism. The research utilizes data deduplication as well as strong encryption to enhance both data security and computational performance particularly for big cloud-based data management during business continuity situations such as pandemic periods. A proposed model utilizes Multi Kernel Fuzzy C-Means (MKFCM) clustering analysis to divide data in accordance with confidence levels before it applies homomorphic encryption to sensitive information and the Optimal SIMON Cipher (OSC) for lightweight optimized security encryption. The implementation of dual encryption along with clustering brings an adverse impact on computational performance that can reduce the scalability and real-time capabilities of high-throughput cloud systems.Recent study proposed by Neve et al. (2024) demonstrates performance analysis along with avalanche evaluation using the hybrid SIMON-SPECK model optimized for efficiency, but lacks consideration of standard ciphers and has limited cloud scalability.

2.3 Comparsion Table

This table 1 summarizes and compares recent research studies on lightweight encryption techniques, particularly for Iot and cloud systems. Each row outlines a study's authors, year, and core technique or focus area—ranging from ARX-based encryption and cryptanalysis to dual encryption and performance evaluation. Key strengths like low memory use, real-time security, and multi-layered protection, are highlighted alongside limitations like lack of attack resistance analysis, complex key management, and limited real-world validation. The table offers a concise overview of current trends, trade-offs, and research gaps in secure, efficient cryptographic methods for resource-constrained environments.

Table 1: Comparative Analysis of Lightweight Encryption Techniques in IoT and Cloud Systems

Author(s)	**Technique / Focus**	**Strengths**	**Weaknesses**
Sleem et al. (2021)	ARX-based SPECK with dynamic key substitution	✓ Reduced round count (26 → 7) ✓ Efficient on IoT hardware	✗ Complex key management
Dhakare et al. (2024)	Performance evaluation in embedded systems	✓ Low memory use ✓ Operational efficiency	✗ No in-depth cryptographic analysis ✗ No attack testing
AbdulRaheem et al. (2022)	Lightweight encryption for edge-IoT healthcare	✓ Real-time security ✓ High speed and confidentiality	✗ Limited evaluation scope ✗ No real clinical deployment
Altaie & Hoomod (2023)	Dual encryption on Raspberry Pi + MQTT	✓ Multi-layered security ✓ Lightweight privacy support	✗ No power or attack analysis ✗ Sparse performance results
Dwivedi & Srivastava (2023)	Heuristic tree-based cryptanalysis of SIMON & SIMECK	✓ Effective differential path detection ✓ Faster analysis	✗ Reduced-round only ✗ Not validated in real-world
Taware et al. (2021)	SIMON with Crow Search Algorithm (CSA) for key optimization	✓ Fast key generation ✓ Maintains secrecy	✗ Retracted publication ✗ Lacks practical validation
Khalifa et al. (2024)	Simulation of sensor-level encryption in IoT	✓ Low CPU usage ✓ Secures low-power devices	✗ Needs optimization ✗ No resistance analysis
Muthumari et al. (2021)	Cloud security with clustering and deduplication	✓ Dual encryption enhances protection ✓ Reduces storage overhead	✗ High computational cost ✗ Scalability concerns
Neve et al. (2024)	Attack analysis on hybrid-SIMON-SPECKey lightweight cryptographic algorithm	✓ Cryptanalysis of hybrid algorithm ✓ Focus on IoT security	✗ Limited scope to hybrid variants ✗ No hardware benchmarking

3 Methodology

The encryption algorithms used in this study are lightweight cryptographic algorithms.

3.1 AES

The Advanced Encryption Standard (AES) is a symmetric block cipher that has become the de facto standard for encryption across government, financial, and commercial applications Sarkar et al. (2024). AES operates as the encryption standard today because NIST implemented it in 2001 to replace old DES after recognizing its enhanced security performance. AES works with blocks of 128 bits while providing users a selection of key abilities that range from 128 to 256 bits to 192 bits. AES performs encryption through

a substitution-permutation network (SPN) built with three round quantity options that build block functions using SubBytes followed by ShiftRows and MixColumns and ends with AddRoundKey. Through AES encryption operations the system implements confusion together with diffusion to protect it against known cryptanalytic attacks including linear and differential cryptanalysis Lavanya and Karpagam (2020). The growing importance of data protection in mobile and cloud and IoT environments keeps AES at the forefront of secure communication tools since the encryption method provides an excellent balance of speed with confidentiality abilities and adaptability features.

3.2 DES

NIST approved the Data Encryption Standard (DES) as a federal standard in 1977 after IBM created it throughout the early 1970s. The Data Encryption Standard performs functional encryption on 64-bit blocking data units using 56-bit keys within 16-round Feistel structure stated by Meça (2023). The DES operating rounds consist of sequential processes which implement expansion and permutation then S-Box substitution and subkey XOR operations that obtain their values from the basic key. The initial breakthrough of DES with federal standardization became obsolete because better computing power eradicated its security entirely. Descriptor attacks now take under one day to decrypt messages protected with DES security because its 56-bit keys are no longer secure. The diminished security of DES enabled developers to build Triple DES (3DES) and AES algorithms for secure encryption methods. The DES protocol serves limited use in contemporary systems but exists mainly as historical reference since modern security standards do not employ it. The encryption method survives through legacy systems though its main role today is historical reference in cryptographic educational materials. The organized structure of DES helps explain basic encrypted cipher structures and binary-level symmetric key operations which led to modern cryptographic development.

3.3 SPECK

The National Security Agency (NSA) developed SPECK in 2013 as a lightweight block cipher Khare (2021) which targets limiting resource devices including IoT devices as well as embedded systems and low-power sensors. Among its ciphers this technology offers the SIMON version for hardware-oriented needs alongside SPECK which performs best in software-based operations. SPECK employs the basic Feistel-like structure with ARX additions (rotation and XOR operations) to maintain efficiency on processors with minimal power and features. SPECK works with diverse combinations of key and block sizes which include SPECK32/64 as well as SPECK128/256 thus providing adaptable security options for various use cases. The simple design of SPECK allows developers to create concise code implementations while running encryption operations and decryption at much faster than AES-type ciphers can handle in resource-limited systems. SPECK operates efficiently yet remains under scrutiny since its development ties back to the NSA while various cryptographic experts debated potential weaknesses or hidden decryption artifacts yet scientists did not find concrete proof of such problems. The current rounds of SPECK stand against both differential and linear cryptanalysis attack methods. SPECK operates as a promising encryption solution within lightweight applications because of its high efficiency combined with flexibility which makes it useful for lightweight cryptography growth.

3.4 SIMON

The National Security Agency introduced the lightweight block cipher SIMON to the market as part of their 2013 SPECK release specifically designed for minimal hardware requirements of RFID tags along with smart cards and embedded systems. SIMON depends on a balanced Feistel network structure implemented with just bitwise logical operations consisting of AND, XOR and circular shifts in order to minimize hardware requirements and reduce power usage. The cipher exists in different versions as SIMON32/64, SIMON64/128, and SIMON128/256 with block size and key size specified by the numbers Ebrahimi et al. (2023). SIMON provides adaptable design features that make it a strong match for various applications needing different security measures and resource constraints. SIMON's round function implements basic cryptographic operations which results in strong diffusion outcomes together with non-linear properties that counteract differential and linear cryptanalytic attacks. The structure of SIMON stands firm after extensive study as experts found no workable assaults against its entire round versions although speculation exists about its transparency similarities to SPECK because it comes from the NSA. The core features of adaptability along with high operational efficiency position SIMON as an industry-leading lightweight encryption system that protects many low-power embedded systems with real-time requirements.

3.5 Parallel Hybrid Encryption and Equal Data Partitioning

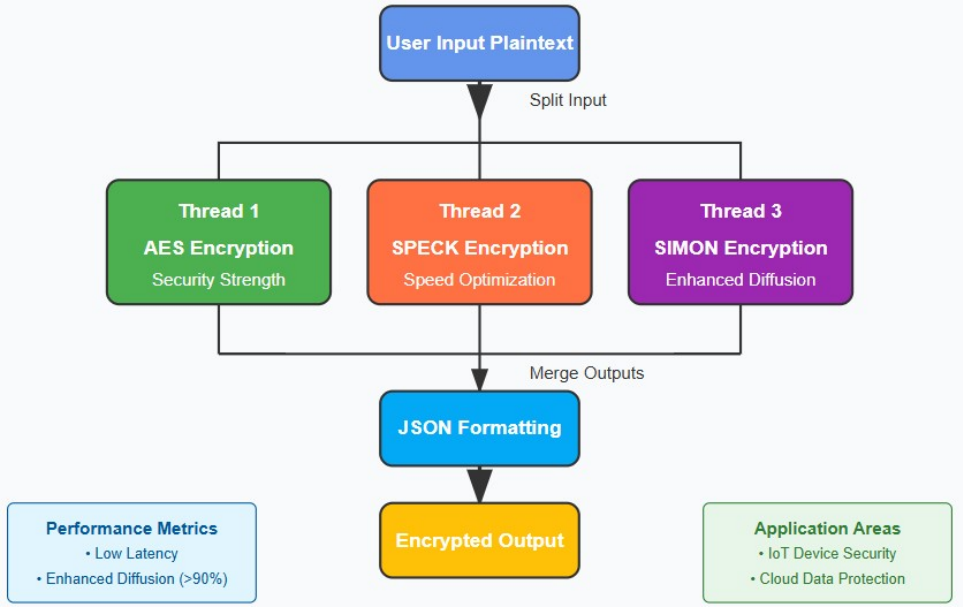

Figure 1: System Workflow Diagram of Parallel-Hybrid Encryption and Equal Data Partitioning

This system workflow diagram in Figure 1 shows the ASSET parallel lightweight cryptographic framework, showing how it processes data to secure IoT and cloud environments.

The core motivation behind adopting a parallel hybrid encryption model lies in achieving an optimal balance between encryption strength, processing latency, and resource efficiency, particularly for IoT and edge computing devices. While it is theoretically possible

to minimize encryption latency to near-zero using extremely lightweight or insecure algorithms, this often leads to weakened encryption strength and poor diffusion properties. In contrast, traditional heavyweight ciphers provide strong security but are unsuitable for low-power devices due to high latency and computational demands. Our approach seeks to bridge this gap by parallelizing the encryption process across three proven cryptographic algorithms—AES, SPECK, and SIMON—where each algorithm contributes uniquely to the overall performance: AES ensures robust security, SPECK adds speed, and SIMON enhances diffusion. To enable efficient parallelization, the plaintext input is divided into three equal parts, each processed simultaneously by one of the algorithms using multi-threading. This division strategy was selected after rigorous experimentation. Initial attempts to divide data into unequal segments introduced significant issues in the decryption process, this led to implementation complexity and threaten decryption consistency particularly in maintaining proper order during reassembly. Uneven segmentation also led to imbalanced thread execution, where one thread would consistently finish earlier or later than others, introducing synchronization overhead and negating the benefits of parallelization. Moreover, by dividing the input equally and ensure simplified reconstruction, and reduced complexity during decryption. The resulting latency from this approach remains within tolerable thresholds for most IoT devices, which are generally capable of handling minimal processing delays as long as security is uncompromised. Therefore, the decision to parallelize the encryption while maintaining equal partition sizes is a deliberate design choice that effectively balances performance, encryption strength, and resource adaptability, making it ideal for cloud-integrated, real-time IoT environments.

4 Design Specification

4.1 Environment Tools

The implementation of the proposed parallel hybrid cryptographic system is carried out using Python, selected for its rich ecosystem of libraries and ease of integration for threading and encryption functionalities. The PyCryptodome library is used to implement the AES algorithm, ensuring robust and secure encryption standards. For SPECK and SIMON, lightweight ciphers developed by the NSA, the system utilizes open-source Python implementations, ensuring optimized performance suitable for constrained devices. The overall encryption and decryption processes are executed using multi-threading to enable parallel processing, significantly reducing encryption latency. The development and testing environment is primarily a PC-based simulation, which allows controlled evaluation of system performance. The system's performance is evaluated using several key metrics. Encryption and decryption latency are measured using Python's timeit module to benchmark the time efficiency of the parallel model against sequential approaches. The avalanche effect, which indicates the diffusion strength of the encryption algorithm, is calculated using Hamming distance, where one-bit changes in the input are analyzed for their impact on the ciphertext. In addition, CPU and RAM utilization are monitored through profiling tools to assess the computational footprint of the model, ensuring it remains lightweight and efficient for IoT environments. Finally there is a comparative analysis which has been conducted between the proposed parallel model and traditional sequential model to validate improvements in speed, diffusion quality and resource optimization.

4.1.1 System Architecture Diagram

In the implementation, the ASSET model outputs a secure encrypted ciphertext by parallelizing three lightweight cryptographic algorithms—AES, SPECK, and SIMON—each operating on a third of the input plaintext message as shown in Figure 1. The solution is built using Python, leveraging the PyCryptodome library for AES and the simonspeckciphers library for SPECK and SIMON. A Flask-based web interface deployed on AWS Cloud9 receives HTTP requests from users or IoT devices. The plaintext is split into three parts, processed concurrently using multi-threading, and then merged after encryption. The final result is encoded in JSON format, which forms the transformed encrypted data output. Implementation avoids code listings or user manual-style documentation and focuses instead on system behavior and result generation. The outcome includes encrypted messages ready for secure transmission, backed by robust encryption logic and performance insights, all generated using open-source cryptographic libraries and AWS cloud infrastructure for real-world applicability.

Figure 2 is showing system architecture diagram of this study.

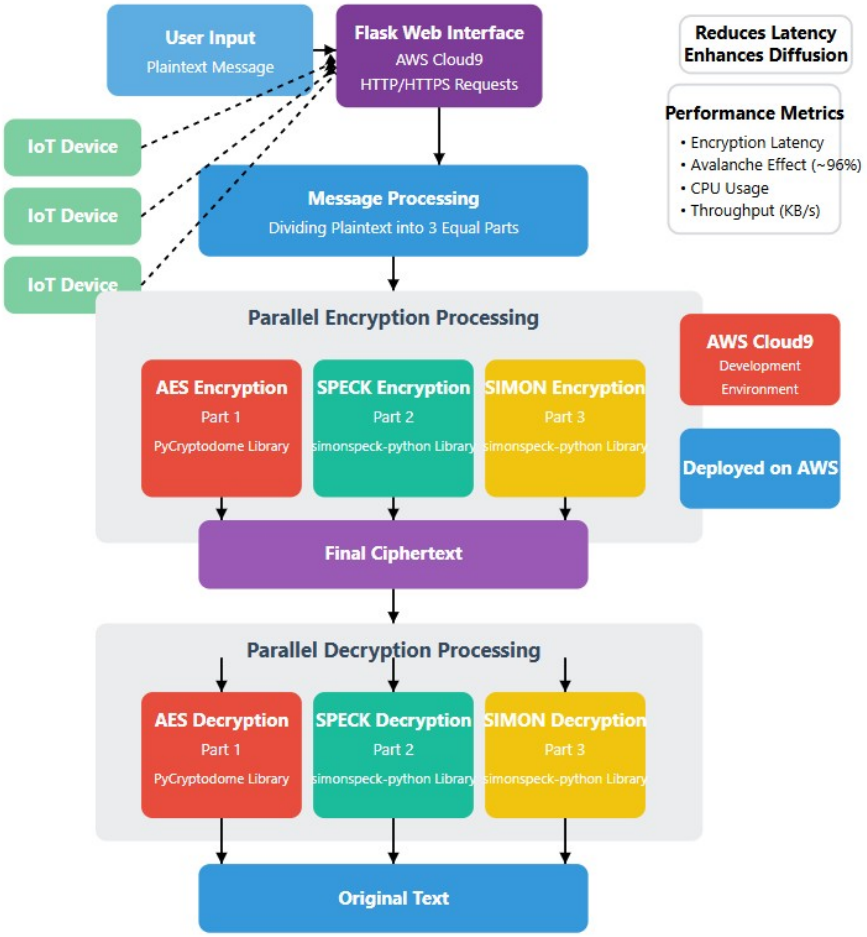

Figure 2: System Architecture Diagram

5 Implementation

5.1 Encryption Process

```
Function ASSET_Encrypt(plaintext, key_AES, key_SPECK, key_SIMON):
  // Step 1: Split the plaintext into 3 equal parts
  [part1, part2, part3] = split_equally(plaintext)
  // Step 2: Define threads for parallel encryption
  Thread T1:
    cipher1 = AES_Encrypt (part1, key_AES)
  Thread T2:
    cipher2 = SPECK_Encrypt (part2, key_SPECK)
  Thread T3:
    cipher3 = SIMON_Encrypt (part3, key_SIMON)
  // Step 3: Start all threads in parallel
  Start T1, T2, T3
  Wait for T1, T2, T3 to finish
  // Step 4: Combine encrypted parts into a final ciphertext (e.g., JSON)
  ciphertext = encode_JSON (cipher1, cipher2, cipher3)
  Return ciphertext
```

Figure 3: ASSET Encryption Algorithm Psuedocode

Figure 3 shows ASSET Encryption Algorithm which encrypts data using a parallel hybrid approach. It splits the plaintext into three equal parts, encrypts each part simultaneously using AES, SPECK and SIMON in separate threads and then combines the encrypted outputs into a single ciphertext. A parallel hybrid cryptographic system's encryption design has been used multi-threaded execution of AES, SPECK together with SIMON to both minimize delays and increase diffusion in the encryption process. After receiving a plaintext message from the user the system automatically splits it into three equal sections. The system distributes three parts to individual encryption threads for parallel execution. AES security gives strong protection due to its widespread use and advanced key management system when encrypting the first part. The encryption of segment two uses SPECK because it delivers high speed and finite hardware efficiency required by real-time IoT systems. SIMON processes the third encryption segment and provides superior diffusion properties that enhance the security features of the complete system. The combined operation of these three encryption threads minimizes overall encryption duration because they run simultaneously rather than in consecutive order. The distinct chunks produced by ASSET encryption run in parallel until JSON formatting turns them into a unified ciphertext that maintains easy decoding potential during decryption. This JSON object uses hexadecimal format to encode encrypted outputs which enables secure standard data handling methods. The parallel execution method

produces a secure and complex output through its use of multiple encryption techniques which protect different parts of the plaintext. This system combines speed with security while achieving diffusion as its primary metrics during data protection thus making it suitable for protecting real-time resources in cloud or Internet of Things environments that need minimal computational requirements.

5.2 Decryption Process

```
Function ASSET_Decrypt (ciphertext, key_AES, key_SPECK, key_SIMON):
  // Step 1: Decode the JSON to retrieve encrypted parts
  [cipher1, cipher2, cipher3] = decode_JSON (ciphertext)
  // Step 2: Define threads for parallel decryption
  Thread T1:
    part1 = AES_Decrypt (cipher1, key_AES)
  Thread T2:
    part2 = SPECK_Decrypt (cipher2, key_SPECK)
  Thread T3:
    part3 = SIMON_Decrypt (cipher3, key_SIMON)
  // Step 3: Start all threads in parallel
  Start T1, T2, T3
  Wait for T1, T2, T3 to finish
  // Step 4: Reassemble the decrypted segments
  plaintext = concatenate (part1, part2, part3)
  Return plaintext
```

Figure 4: ASSET Decryption Algorithm Psuedocode

Figure 4 shows ASSET Decryption Algorithm where decryption process reverses the parallel encryption. It decodes the combined ciphertext which extracts the encrypted segments and decrypts each in parallel.

A parallel decryption strategy provides the method to obtain original plaintext by matching the encryption format during the decryption process. The decryption process starts by processing JSON-encoded ciphertext that contains separately encrypted ASSET outputs. The JSON object processing reveals each cipher's output data through decoding its hexadecimal format. The decoded encrypted segments move to three threads that execute separate decryption operations in parallel. The AES decryption operation takes place in the initial thread by using the same decryption key and mode that was used during encryption while processing its allocated encrypted segment. The decryption process using SPECK function operates on its specific segments by ensuring powerful lightweight processing takes place. The final encryption portion goes through SIMON decryption

processing within the third thread to achieve its original state. Three executing decryption functions with multi-threading reduces the typical latency delays occurring in layered decryption methods. After decryption of all three segments succeeds the system reassembles the plaintext message by reconnecting its parts in their original order. Encryption formatting along with padding elements must be removed for the final output to align with the original user content. The parallel decryption method protects both system speed and encryption strength by observing security rules of each separate cipher. The combined system architecture provides efficient data security along with rapid accessibility which makes it highly useful for real-time IoT and cloud applications demanding strong diffusion and low latency.

6 Evaluation

The results of the ASSET model underscore a critical trade-off between security, performance, and computational efficiency, which this study successfully navigates. By integrating AES, SPECK, and SIMON in a parallelized architecture, the model achieves a synergistic balance—delivering faster encryption and decryption times without compromising security. While SPECK offers exceptional speed and SIMON enhances diffusion, AES ensures robust cryptographic strength, allowing the hybrid approach to outperform traditional lightweight models such as LWC-AES in both avalanche effect and throughput. Although multi-threading introduces minor synchronization overhead, this is outweighed by the significant reduction in latency and improved resource distribution across threads. The equal partitioning strategy ensures consistent execution and simplified reassembly, validating the system's reliability for real-time IoT and cloud environments. Thus, the ASSET framework demonstrates that a carefully engineered parallel hybrid model can offer an optimal trade-off—maintaining strong encryption standards while addressing latency and efficiency in constrained systems.

6.1 Case Study 1: AES + SPECK + SIMON Encryption Technique (ASSET)

The performance evaluation of the proposed ASSET is demonstrated through a case study focusing on byte-wise throughput measured in kilobytes per second (KB/s). The results illustrate how the system performs when encrypting data of increasing byte sizes, from 1 to 5 bytes. For a 1-byte message, the throughput is notably high at 3149.764 KB/s, indicating exceptional performance for minimal data sizes, which is highly beneficial in IoT systems that frequently transmit small data packets. With 2 bytes, the throughput peaks at 4548.404 KB/s, suggesting the model's performance is optimized when handling slightly larger payloads in parallel. However, at 3 bytes, a drop in throughput is observed, reducing to 2669.595 KB/s, potentially due to thread synchronization overhead or computational balancing between the ciphers. For 4 and 5 bytes, throughput stabilizes at 2706.127 KB/s and 2851.847 KB/s as shown in Figure 5 which shows the model's ability to maintain high performance even as message size increases. The ASSET model showed consistently high throughput across varying byte sizes, validating its suitability for real-time applications in constrained environments. This case study confirms that the parallel cryptographic approach significantly reduces processing delays while maintaining lightweight efficiency and enhanced diffusion. The avalanche score

evaluation of the ASSET shows the system's strong diffusion capability across varying input sizes. The avalanche effect, measured as a percentage of bit changes in the ciphertext when a single input bit is altered, remains consistently high across all tested byte sizes. For 1 and 5 bytes, the model achieves a peak score of 96.093 percentage in Figure 6 showing excellent diffusion. Scores for 2, 3, and 4 bytes are 91.406, 94.531, and 92.96 percentage respectively, showing minimal variation and confirming the model's reliability in producing highly unpredictable ciphertext, crucial for cryptographic security.

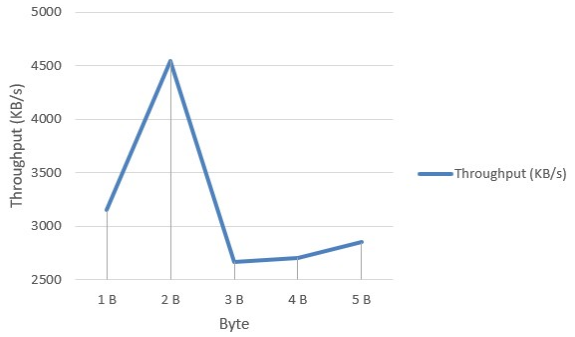

Figure 5: Throughput vs Bytes

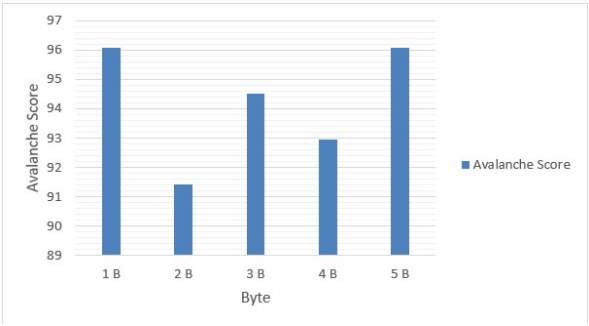

Figure 6: Avalanche Score vs Bytes

Table 2: Evaluation of AES, SIMON and SPECK

Cryptographic Algorithm ASSET	**Encryption Time in ms**	**Decryption Time in ms**
9B	11.8	5.72
11B	6.77	18.19
20B	12	37.96

6.2 Case Study 2: DES + SPECK + SIMON Encryption Technique (DSSET)

The DSSET demonstrates varying throughput performance across different byte sizes. At 1 byte, the throughput is exceptionally high at 13658.311 KB/s which is been likely due to DES's faster block processing in minimal data scenarios. However, as the input size increases, the throughput fluctuates dropping to 4956.093 KB/s at 2 bytes, rising slightly at 3 bytes (5763.135 KB/s), and dipping again at 4 bytes (2861.242 KB/s) as shown in Figure 7. A notable increase is observed at 5 bytes with 6200.494 KB/s.

Table 3: Evaluation of DES, SIMON and SPECK

Cryptographic Algorithm DSSET	**Encryption Time in ms**	**Decryption Time in ms**
9B	17.2	14.62
11B	30.35	20.14
20B	25.73	66.2

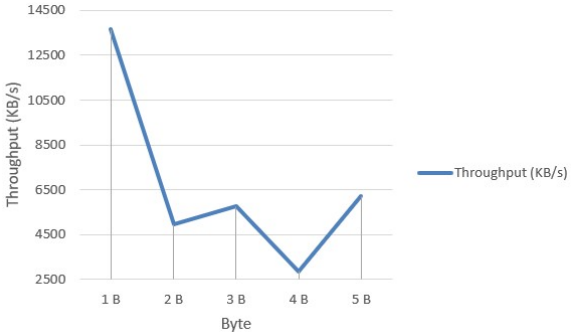

Figure 7: Throughput vs Bytes)

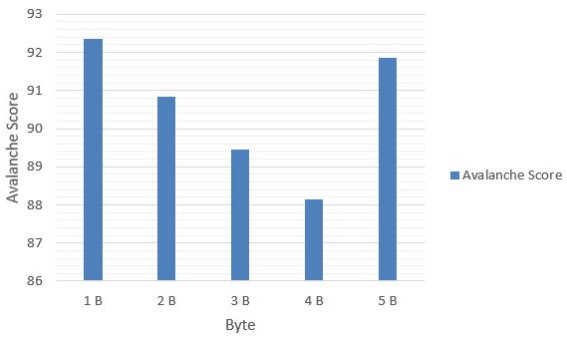

Figure 8: Avalanche Score vs Bytes

The avalanche score results for the DSSET indicate a moderately strong but slightly less consistent diffusion effect compared to the ASSET model. As the input size increases from 1 to 5 bytes, the avalanche scores range from 88.138 to 92.361 percentage which reflects a generally high but more fluctuating diffusion performance as shown in Figure 8. The highest score, 92.361 percentage, occurs at 1 byte, while the lowest, 88.138 percentage, is observed at 4 bytes. Although DSSET maintains acceptable cryptographic strength, the scores suggest slightly reduced diffusion compared to ASSET making it relatively less optimal for applications demanding maximum unpredictability and security.

6.3 Comparison of Model Performance: Previous Study vs. Current Study

Figure 9 compares encryption time vs input size across three cryptographic models. LWC-AES shows significantly higher encryption times (289-712ms) compared to the parallel hybrid models ASSET (6.77-12ms) and DSSET (17.2-30.35ms). As input size increases from 9 to 20 bytes, LWC-AES performance improves dramatically, dropping from 712ms to 289ms, while the parallel models maintain consistently low encryption times across all input sizes. This demonstrates that the parallel hybrid approaches achieve substantially better performance than the LWC-AES modification from the previous research given by Mohammad and Abdullah (2022).

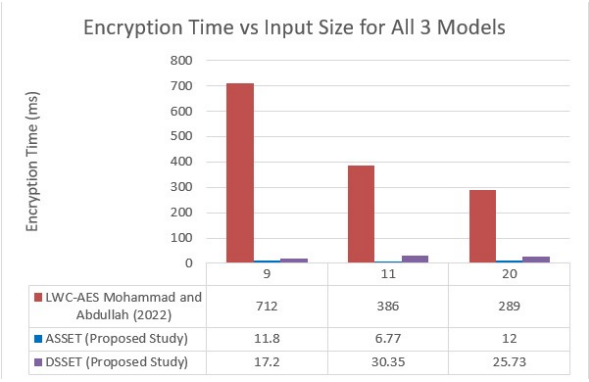

Figure 9: Comparison of Encryption Time of previous study vs current study models

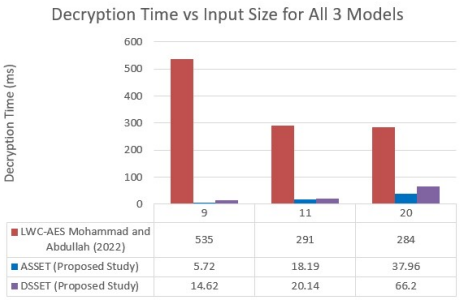

Figure 10: Comparison of Decryption Time of previous study vs current study models

Figure 10 compares decryption time vs input size across three cryptographic models. LWC-AES from the previous study Mohammad and Abdullah (2022) shows substantially higher decryption times (284-535ms) compared to our parallel hybrid models ASSET

(5.72-37.96ms) and DSSET (14.62-66.2ms). As input size increases from 9 to 20 units, LWC-AES performance improves significantly, dropping from 535ms to 284ms, while our parallel models show a gradual increase in decryption time with larger inputs. Despite this trend, our parallel approaches maintain dramatically better performance than the lightweight AES modification from the previous research Mohammad and Abdullah (2022) with ASSET delivering the fastest decryption for smaller inputs and maintaining strong performance across all input sizes.

6.4 Comparsion Table

Table 4: Comparison of Encryption Models: LWC-AES vs ASSET and DSSET

Features	**Base Paper: LWC-AES (Mohammad & Abdullah, 2022)**	**ASSET (Proposed Study)**	**DSSET (Proposed Study)**
Core Algorithm	Lightweight AES (AES without MixColumns + Continued Fraction)	AES + SPECK + SIMON (parallel)	DES + SPECK + SIMON (parallel)
Execution Strategy	Single-threaded (Sequential AES)	Multithreaded Parallel Encryption	Multithreaded Parallel Encryption
Platform & Tech Stack	ASP.NET, C#, Windows 10	Python, Flask, AWS Cloud9	Python, Flask, AWS Cloud9
Encryption Time (ms)	712 (9B), 386 (11B), 289 (20B)	11.8, 6.77, 12	17.2, 30.35, 25.73
Decryption Time (ms)	535 (9B), 291 (11B), 284 (20B)	5.72, 18.19, 37.96	14.62, 20.14, 66.2
Security Enhancement	Partial (Removed MixColumns may weaken resistance)	Stronger via hybrid multilayer encryption paths	Stronger via hybrid multilayer encryption paths
Avalanche Effect Tested	Yes (52%)	Around 96%	Around 90%
Usability in Cloud/Web	Desktop-based, no API/Web integration	Cloud-ready Web app + APIs deployed on AWS	Cloud-ready Web app + APIs deployed on AWS
Application Domain	Low-resource devices (IoT, smartcards)	Cloud platforms, IoT APIs, and responsive web applications	Cloud platforms, IoT APIs, and responsive web applications

The Table 4 explains the comparison among my Proposed Models and LWC-AES given by Mohammad and Abdullah (2022) shows the comparison of evaluation results and demonstrates the trade-off between cryptographic strength and lightweight efficiency with respect to the base paper and the proposed study. The base paper optimizes the standard AES algorithm by using continued fraction which is a mathematical function and removes MixCloumn method to reduce complexity and improve encryption speed but compromises diffusion with low avalanche score of 52 percent and shows limited scalability. The proposed study developed parallel-based hybrid encryption models AES/SPECK /SI-

MON (ASSET) and DES/SIMON/SPECK(DSSET) with significantly improved diffusion results of around 96 percent and 90 percent avalanche effect and throughput (1607.39, 5379.12 kb/sec) respectively along with drastically reduced latency, which proves the balance between encryption strength and diffusion makes them suitable for IOT devices with tolerable latency.Both the models have relatively low resource consumption, whereas the DSSET model has slightly higher CPU usage than ASSET model. In the proposed study ASSET model offers stronger security and faster processing with minimal computational cost makes it suitable for real-time, cloud integrated environments.

7 Conclusion and Future Work

7.1 Conclusion

This security project develops a new hybrid cryptography method which unites AES together with SPECK and SIMON ciphers to provide secure and efficient security solutions for cloud technology.The encryption algorithms were chosen due to their complementary cryptographic structure and operational efficiency. The ASSET model improves both security strength and speed capabilities by merging AES encryption with SPECK encryption yet adopting SIMON to optimize diffusion. This parallel execution method allows message segments to be encrypted concurrently which reduces processing time instead of creating additional delays like sequential layered encryption. This allows the model to maintain lightweight performance without compromising on cryptographic protection. The experimental results showed that the ASSET model provided superior performance than traditional methods in terms of encryption speed and decryption speed and system resource usage and diffusion strength capabilities. Real-time simulations showed that the system achieved high throughputs as well as avalanche scores which exceeded 90 percentage thus showing excellent resilience to differential cryptanalysis attacks. The system operation demonstrated optimal usage of CPU and memory thus proving it can run in real-time applications within embedded and restricted deployments. This research presents a secure cryptographic solution that demonstrates scalability for protecting IoT-based communication networks. The system demonstrates potential deployment potential through the implementation of a Flask web interface that demonstrates its utility as a lightweight cryptographic microservice available for cloud-based or edge computing platforms. The proposed research advances by integrating both standard and lightweight ciphers in parallel. This approach makes this model unique and focus on enhancing the overall performance, encryption strength and security along with cloud deployment scalability. The parallel hybrid implementation demonstrates success as a suitable encryption method because it provides enhanced security protection combined with compatibility for contemporary IoT requirements during deployments in Raspberry Pi, ESP32 and STM32 microcontroller systems.

7.2 Future Works

While the ASSET model has demonstrated enhanced encryption performance in terms of speed, diffusion, and resource efficiency, future directions aim to extend its applicability to cloud environments. This encryption technique holds promise for securing cloud-specific resources such as database snapshots, AMI (Amazon Machine Images), and objects stored in S3 buckets. From a cloud perspective, the ASSET model can be adapted to encrypt

data-at-rest within storage services or data-in-transit across distributed systems. Its lightweight and parallelized structure makes it suitable for integrating into cloud-based microservices or serverless functions, where performance and latency are critical. Future implementations could explore the encryption of cloud-native objects and validate the model's utility in real-time cloud deployments. Justifying its effectiveness in terms of improved performance and enhanced security would further strengthen its case for adoption. Moreover, leveraging this technique for betterment of cloud services—such as data privacy, integrity, and compliance—can transform it into a comprehensive solution for modern cloud security needs.

References

AbdulRaheem, M., Oladipo, I. D., González-Briones, A., Awotunde, J. B., Tomori, A. R. and Jimoh, R. G. (2022). *An efficient lightweight speck technique for edge-IoT-based smart healthcare systems*, Academic Press, pp. 139–162.

Ahmad, N., Zamri, M. H., Ahmed, S., Wijayanto, A. and Isaak, S. (2024). Design of advanced encryption system (aes) algorithm using asic implementation for internet of things (iot) devices, *Journal of Advanced Research Design* **123**(1): 131–145.

Al-Mashhadani, M. and Shujaa, M. (2022). Iot security using aes encryption technology based esp32 platform, *Int. Arab J. Inf. Technol.* **19**(2): 214–223.

Altaie, R. H. and Hoomod, H. K. (2023). Hybrid speck encryption algorithm for internet of thing (iot), *International Conference of Reliable Information and Communication Technology*, Springer Nature Switzerland, pp. 317–326.

Arpaia, P., Bonavolonta, F. and Cioffi, A. (2020). Problems of the advanced encryption standard in protecting internet of things sensor networks, *Measurement* **161**: 107853.

Chahar, S. (2025). *Exploring the future trends of cryptography*, CRC Press, pp. 234–257.

Dhakare, S., Chippalkatti, S. S. and Misbahuddin, M. (2024). Securing the iot device network with lightweight cryptography, *2024 27th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, IEEE, pp. 1–5.

Dwivedi, A. D. and Srivastava, G. (2023). Security analysis of lightweight iot encryption algorithms: Simon and simeck, *Internet of Things* **21**: 100677.

Ebrahimi, A., Gerault, D. and Palmieri, P. (2023). Deep learning-based rotational-xor distinguishers for and-rx block ciphers: Evaluations on simeck and simon, *International Conference on Selected Areas in Cryptography*, Springer Nature Switzerland, pp. 429–450.

Khalifa, O. O., Ahmed, M. Z. and Hashim, A. H. A. (2024). Application of simon encryption algorithm for data transmission between sensor nodes in iot environment, *2024 9th International Conference on Mechatronics Engineering (ICOM)*, IEEE, pp. 127–132.

Khare, S. (2021). *Efficient and Enhanced Lightweight Hybrid Cryptosystem for Class-0 IoT Data Security Using Elliptic Curve Cryptography and Speck*, PhD thesis, University of Louisiana at Lafayette.

Lavanya, R. and Karpagam, M. (2020). Enhancing the security of aes through small scale confusion operations for data communication, *Microprocessors and Microsystems* **75**: 103041.

Meça, A. (2023). Exploring data encryption standard (des) through cryptool implementation: A comprehensive examination and historical perspective, *International Conference for Emerging Technologies in Computing*, Springer Nature Switzerland, pp. 143–160.

Mohammad, H. M. and Abdullah, A. A. (2022). Enhancement process of aes: a lightweight cryptography algorithm-aes for constrained devices, *TELKOMNIKA (Telecommunication Computing Electronics and Control)* **20**(3): 551–560.

Muthumari, A., Banumathi, J., Rajasekaran, S., Vijayakarthik, P., Shankar, K., Pustokhina, I. V. and Pustokhin, D. A. (2021). High security for de-duplicated big data using optimal simon cipher, *Computers, Materials & Continua* **67**(2).

Naser, S. M. (2021). Cryptography: from the ancient history to now, it's applications and a new complete numerical model, *International journal of mathematics and statistics studies* **9**(3): 11–30.

Salman, R. S., Farhan, A. K. and Shakir, A. (2022). Lightweight modifications in the advanced encryption standard (aes) for iot applications: a comparative survey, *2022 International Conference on Computer Science and Software Engineering (CSASE)*, IEEE, pp. 325–330.

Sarkar, B., Saha, A., Dutta, D., De Sarkar, G. and Karmakar, K. (2024). A survey on the advanced encryption standard (aes): A pillar of modern cryptography. Preprint.

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* pp. 1–18.

Sleem, L. and Couturier, R. (2021). Speck-r: An ultra light-weight cryptographic scheme for internet of things, *Multimedia Tools and Applications* **80**(11): 17067–17102.

Sultan, I., Mir, B. J. and Banday, M. T. (2020). Analysis and optimization of advanced encryption standard for the internet of things, *2020 7th International Conference on Signal Processing and Integrated Networks (SPIN)*, IEEE, pp. 571–575.

Taware, S., Chakravarthi, R. R., Palagan, C. A., Chandrasekaran, K. and Vadivelan, N. (2021). Preserving mobile commerce iot data using light weight simon block cipher cryptographic paradigm, *Journal of Ambient Intelligence and Humanized Computing* **12**: 6081–6089.