

Authentication of IOT devices using PUF and Encryption Technique

M.Tech Thesis

*Submitted in partial fulfillment of
the requirements for the award of the degree
of*

Master of Technology

in

Department of Software Engineering

submitted by

Mohit Sharma (23/SWE/16)

under the guidance of

Dr. Sanjay Patidar

Assistant Professor

Department of Software Engineering



DEPTT. OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY, DELHI

May 2025

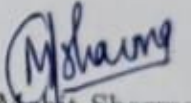
DEPARTMENT OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I am very thankful to Dr Sanjay Patidar (Assistant Professor, Department of Software Engineering) and all the Department of Software Engineering faculty members at DTU. They all provided me with immense support and guidance for the project.

I would also like to express my gratitude to the University for providing me with the laboratories, infrastructure, testing facilities, and environment, which allowed me to work without obstructions.

I would also like to express my appreciation for the support provided to me by the lab assistants, seniors, and our peer group, who aided me with all the knowledge they had regarding various topics.


Mohit Sharma

(23/SWE/16)

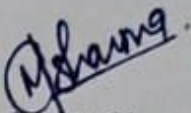
DEPARTMENT OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I hereby declare that the M.Tech Thesis entitled **Authentication of IOT devices using PUF and Encryption Technique**, which is being submitted to Delhi Technological University, in partial fulfilment of requirements for the award of the degree of Master Of Technology (Software Engineering) is a bonafide report of M.Tech Thesis carried out by me. The material contained in the thesis has not been submitted to any university or institution for the award of any degree.

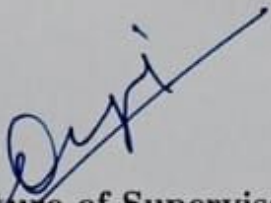
Date: 19 June 2025

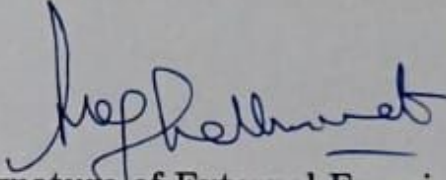
Place: New Delhi


Mohit Sharma

23/SWE/16

This is to certify that the student has incorporated all the corrections suggested by the examiner in the thesis and the statement made by the candidate is correct to the best of our knowledge. .


Signature of Supervisor


Signature of External Examiner

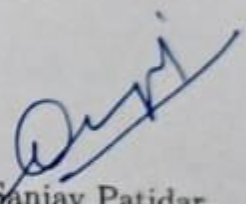
DEPARTMENT OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

This is to certify that M.Tech Thesis entitled **Authentication of IoT Devices Using PUF and Encryption Technique** which is submitted by Mohit Sharma, Roll No - 23/SWE/16, Department of Software Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of degree Master Of Technology (Software Engineering) is a record of the candidate work carried out by him under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Date: 19 June 2025

Place: New Delhi


Dr. Sanjay Patidar
Assistant Professor
Department of Software Engineering
Delhi Technological University

ABSTRACT

The rapid proliferation of the Internet of Things (IoT) has introduced unprecedented opportunities for automation, intelligence, and connectivity across various domains, including healthcare, smart cities, industrial systems, and personal environments. However, the security and privacy challenges associated with such interconnected systems remain critical concerns, especially given the constrained computational resources and heterogeneous nature of IoT devices. This review synthesizes and critically analyzes ten seminal research contributions addressing various dimensions of IoT security. The focus areas include lightweight cryptographic algorithms, preference-based privacy protection, secure authentication and access control mechanisms, identity management frameworks, and the integration of emerging technologies such as Physical Unclonable Functions (PUFs) and Software Defined Networking (SDN). Additionally, the survey covers architectural perspectives, enabling technologies, and implementation challenges pertaining to IoT security models. Each paper is dissected in terms of its methodology, technical innovation, and effectiveness in mitigating specific threats such as identity theft, unauthorized access, data leakage, and network-level intrusions. The review also provides a comparative evaluation of these approaches, highlighting their strengths, limitations, and applicability to different IoT environments. A dedicated chapter presents essential background concepts to aid reader comprehension, followed by an in-depth discussion on methodological frameworks employed across the literature. Where applicable, implementation insights and experimental outcomes are explored to bridge theory with practice. The final sections distill key findings, identify gaps, and suggest future research directions aimed at enhancing the resilience and scalability of IoT systems. This review is an in-depth resource for practitioners and scholars who wish to comprehend and develop the role of security in the future IoT scenario.

Contents

| | |
|-------------------------------------------------------------------|-------------|
| Acknowledgment | i |
| Declaration | ii |
| Certificate | iii |
| Abstract | iv |
| Contents | vi |
| List of Tables | vii |
| List of Figures | viii |
| 1 Introduction | 1 |
| 1.1 Overview | 1 |
| 1.2 Background | 1 |
| 1.3 Motivation | 2 |
| 1.4 Problem Statement | 2 |
| 1.5 Objectives | 3 |
| 1.6 Technology Used | 3 |
| 1.7 Thesis Structure | 4 |
| 2 Literature Review | 6 |
| 3 Methodology | 12 |
| 3.1 Physically Unclonable Function (PUF) | 12 |
| 3.1.1 Working Principle of PUF | 12 |
| 3.2 PUF-Based Authentication | 13 |
| 3.3 PUF-Based Encryption and Decryption | 15 |
| 3.3.1 PUF-based Key Generation | 15 |
| 3.3.2 Encryption Process | 15 |
| 3.3.3 Decryption Process | 16 |
| 3.3.4 Advantages of PUF-Based Encryption and Decryption | 17 |
| 3.4 Lightweight RC5 Algorithm Encryption | 18 |
| 3.4.1 RC5 Key Expansion | 18 |
| 3.4.2 Encryption Process | 18 |
| 3.4.3 Decryption Process | 20 |
| 3.4.4 RC5 Encryption Parameters | 20 |

| | | |
|----------|--------------------------------------------------|-----------|
| 4 | Experiments and Results | 22 |
| 4.1 | Introduction | 22 |
| 4.2 | Experimental Setup | 22 |
| 4.3 | PUF Authentication Results | 22 |
| 4.4 | Performance Analysis of RC5 Encryption | 23 |
| 4.5 | Comparison with Existing Techniques | 24 |
| 5 | Discussion and Analysis | 25 |
| 6 | Conclusion and Future Work | 28 |
| 6.1 | Conclusion | 28 |
| 6.2 | Future Work | 28 |

List of Tables

| | | |
|-----|-----------------------------------------------------------------------------------------------------------|----|
| 2.1 | Comparison of IoT Security Approaches in Reviewed Papers | 10 |
| 3.1 | RC5 Encryption Parameters | 21 |
| 4.1 | Computational Performance of RC5 Encryption | 23 |
| 4.2 | Comparison with Existing Lightweight Encryption Techniques | 24 |
| 5.1 | Detailed Efficiency Comparison of PUF Authentication, RC5 Encryption, and Combined Framework | 26 |

List of Figures

| | | |
|-----|---------------------------------------------------------------------|----|
| 3.1 | PUF Circuit | 13 |
| 3.2 | PUF-Based Authentication Workflow Between IoT Device and Server . . | 14 |
| 3.3 | PUF Based Encryption Scheme | 16 |
| 3.4 | PUF-Based Decryption Scheme | 17 |
| 3.5 | Flow Diagram Based on RC5 and PUF | 19 |
| 5.1 | Comparison of Latency, Memory CPU and Energy | 27 |
| 5.2 | Efficiency Comparison | 27 |

Chapter 1

Introduction

1.1 Overview

Device communication, interaction, and information processing are all significantly changing with the Internet of Things (IoT). It is a process in which ordinary physical objects are connected to the internet in order to allow them to receive and exchange information. IoT has applications from smart homes and healthcare systems to smart cities and industrial automation. With the ever-growing network reaching billions of devices that connect, privacy and security of the connected systems become increasingly challenging.

In order to safeguard user information and preserve system integrity, security in the Internet of Things is not only a technological need but also an essential one. As sensors, actuators, and technology for communication are used more often, risks including theft of identities, breaches of data, and Unauthorized access and intrusions have grown in frequency. Consequently, creating effective and portable safety measures. Research has shifted to a major priority area: security measures appropriate for devices with minimal resources.

This paper examines many approaches to enhancing IoT security that have been put up in the literature. It encompasses methods including hardware-level security, privacy-aware platforms, identity management systems, processes for authentication, and lightweight encryption. These ideas seek to tackle current and novel safety concerns in various IoT settings.

1.2 Background

Devices that make up Internet of Things systems frequently have low amounts of memory, computing power, and the battery's lifespan. Even if they are safe, traditional cryptographic methods might not work well in these kinds of limited settings. For example, the RC5 algorithm was created as a symmetric cipher that is very efficient, simple to use, and memory-friendly. It can be optimized for different hardware requirements, making it an ideal choice for lightweight encryption for Internet of Things products[1].

When sensitive personal information is collected by technologies, privacy problems are also generated. In order to give consumers more control over sharing and processing their data, preference-based policies of privacy protection are being developed [2]. Additionally, in order to ensure secure connection and data transfer between devices, identity management must be integrated into IoT systems [3]. To keep unauthorized usage at bay,

authentication access controls need to be in place. Some of these methods like role-based access and elliptic curve cryptography have been explored for gaining secure but real procedures of authentication [2]. Hardware-based techniques like Physically Unclonable Functions (PUFs) [3] is one more feasible method of authenticating gadgets without storing cryptographic keys in memory. Complexity in creating and maintaining a network grows with the size of the network. Advanced technologies such as Software Defined Networking (SDN) have been utilized for dynamically applying network-level security policies. IoT architecture and enabling technologies analysis focus on creating secure and scalable infrastructure. All these initiatives aim at creating a long-lasting IoT ecosystem in which security is integrated at each level.

1.3 Motivation

The rapid development of IoT technologies for consumer, industrial, and public applications has raised the demand for safe communication and data security to a great extent. As devices are being deployed more under IoT, the threats of hacking are also on the rise. The devices are mostly vulnerable to open environments and possess minimal processing power, so it is very easy for security attacks like unauthorized access, identity theft, and data manipulation [4].

Moreover, the majority of IoT applications involve personal and confidential information, particularly in areas of healthcare, smart home, and smart transport. Preserving data integrity and privacy is essential to providing end-user trust and overall dependability of IoT systems. Lightweight and scalable security mechanisms are needed to match the resource constraints of IoT nodes without compromising safety and privacy [3].

Researchers have proposed various solutions, including cryptographic algorithms, authentication protocols, identity frameworks, and hardware-based security primitives like PUFs. Exploring these solutions helps identify best practices and research gaps in securing IoT infrastructure.

1.4 Problem Statement

IoT has accelerated in the modern world, and there are currently trillions of IoT-enabled gadgets available for purchase. They are widely used for monitoring, research, education, business, health, and security purposes. Even though the Internet of Things is made up of sensors, internet-enabled devices, servers, databases, and online portals, the primary method of gathering data is through sensors that are placed in strategic locations that are easily accessible to hackers. These sensors send data to a device with internet access, which then sends it to a server and database system for processing and storing.

The issue here is that an intruder can get to the sensors and change or swap them out. They can also get crucial information and keep an eye on traffic. These actions may result in an IoT security breach and significant losses. According to several published publications in recent years, privacy and security in IOT have been identified as one of the most difficult topics [4].

PUF (physically unclonable function) and lightweight encryption of transferred traffic are the two components of the solution I'm putting forth.

1.5 Objectives

The main objectives of this review are:

- To analyze the key security challenges in IoT systems.
- To study and compare various research-based solutions for authentication, encryption, and privacy protection in IoT.
- To identify the strengths and limitations of each approach in terms of efficiency, scalability, and applicability.
- To emphasize the significance of lightweight and hardware-supported security models for resource-constrained IoT environments.
- To give a systematic comparison of assessed methodologies and to make recommendations for directions in future research.

1.6 Technology Used

This dissertation covers a range of state-of-the-art technologies applied in Internet of Things (IoT) ecosystems security. The focus technologies are cryptographic protocols, identity management systems, Physically Unclonable Functions (PUFs), Software Defined Networking (SDN), and privacy-preserving architectures.

Cryptographic Algorithms

Cryptographic protocols, specifically those designed for low-resource environments, are the backbone of IoT security. The thesis makes mention of the RC5 symmetric cipher algorithm, which is well known for consuming very low computational and memory resources, and is suitable for low resource devices. Elliptic Curve Cryptography (ECC) [5], being of high security with comparatively small keys, is also mentioned for secure key exchange as well as authentication in IoT networks. Such algorithms guarantee effective encryption along with IoT device security requirements and energy usage.

Identity Management Systems

Cloud-based identity and access management tools [6] are explored as a way to manage IoT security at scale. These tools tie together device identities and access credentials in a manner that allows them to securely authenticate and authorize devices in large-scale IoT environments. The technology is centered on the unification of devices into a secure, scalable cloud framework, where identity authentication, access control, and auditing are tied together.

Physically Unclonable Functions (PUFs)

PUFs [3] are introduced as hardware security primitives offering a key storage-free alternative to device authentication. PUFs take advantage of inherent silicon fabrication variations to generate device-specific unique identifiers, which renders them very hard to clone or modify. This innovation offers a low-power and cheap alternative to conventional cryptographic key storage, especially for resource-limited IoT devices.

Software Defined Networking (SDN)

SDN [4] is introduced as a smart networking model that enables centralized management of network traffic. Separation of the data plane and control plane, SDN enables real-time visibility into the network traffic, allowing dynamic policy-based security to be implemented. It comes handy in IoT, where numerous devices generate massive volumes of data that need to be transferred and processed accurately. SDN facilitates the implementation of intrusion detection systems and automated threat containment mechanisms with ease.

Privacy-Preserving Frameworks

Privacy-protecting technology is crucial in IoT, where confidential data is often exchanged among devices and servers. Tao and Wang’s privacy protection model based on preference [5] is shown as a method to expand user control by providing users with the ability to choose how their data gets disseminated according to pre-defined privacy preferences. This technology values user-oriented data handling with versatility and authority while ensuring that data is protected according to individual preferences.

These technologies together constitute the crux of security solutions developed in this thesis, each solving unique challenges found in the IoT world. The integration of lightweight cryptographic solutions, identity management security, hardware security, and real-time network monitoring provides an exhaustive solution for IoT system security.

1.7 Thesis Structure

This thesis comprises a number of elaborate chapters with the objective to present a comprehensive overview of security issues and solutions of the Internet of Things (IoT). The book starts with an **Abstract** of the research scope, primary findings, and contributions. The **Introduction** chapter provides an overview of IoT technologies, their importance, and the urgent security and privacy issues that render this book necessary. It also outlines the scope, objectives, and the methodological approach of the review. The **Literature Review** chapter presents an extensive examination of ten seminal research papers, critically analyzing various cryptographic methods, authentication protocols, identity management frameworks, hardware-based security primitives such as Physically Unclonable Functions (PUFs), and Software Defined Networking (SDN) approaches tailored for IoT environments. Following this, the **Prerequisites** chapter provides essential background knowledge, including fundamental cryptographic techniques, authentication and access control models, identity management concepts, and emerging technologies like SDN and PUFs, ensuring that readers are well-equipped to understand the subsequent discussions. The **Methodology** chapter delves deeply into the technical mechanisms employed by each reviewed study, dissecting algorithms, architectural designs, protocol workflows, and hardware implementations, with particular emphasis on resource constraints, security features, and scalability. Where applicable, an **Implementation Details** chapter discusses practical considerations, system setups, and validation procedures from experimental or simulation-based evaluations. The **Results and Discussion** chapter synthesizes findings across the reviewed literature, comparing performance metrics, threat resistance, and deployment challenges, while highlighting design trade-offs and emerging trends. The inadequacies and unresolved difficulties with existing solutions, such as those pertaining to

flexibility, interoperability, usability, and changing threat environments, are thoroughly addressed in a special **Limitations** chapter. Lastly, a **Conclusion and Future Work** portion wraps up the thesis by summarizing important findings and suggesting tactical paths forward for IoT security research, such as incorporating adaptive models, hardware improvements, standardized assessment frameworks, and interdisciplinary cooperation. This well-organized design guarantees a smooth transition from basic ideas to more complex analysis, promoting a thorough comprehension of IoT security in both academic and real-world settings.

Chapter 2

Literature Review

The Internet of Things (IoT) interlinks trillions of devices with secure opportunities as well as catastrophic security challenges. In this chapter, several research papers are proposed to present suggested solutions that mitigate these challenges from different aspects like cryptographic approaches, identity management, authentication techniques, software-defined networks, and hardware-based security primitives. The studies altogether support understanding in general how resource-limited IoT environments can be made secure.

Lightweight Cryptography for IoT Devices

One very simple and powerful symmetric block cipher is Rivest's algorithm RC5. Due to its ability to support word sizes, key sizes, and rounds of various kinds, it can be used in different applications, such as IoT devices [6]. RC5 has been suggested to be deployed on low-power embedded devices due to the memory requirement and the data-dependent rotations. Nevertheless, because of the energy needed for writing operations, RC5 may not always function dependably on extremely low power RFID systems and is susceptible to certain timing-based attacks [7].

Tao and Wang propose a privacy framework that considers user preferences while protecting personal data in IoT applications [8]. This work acknowledges that privacy needs vary across users and contexts. The proposed solution enables dynamic access control based on predefined preferences, ensuring flexible data sharing while respecting user concerns. The mechanism leverages a context-aware model to determine privacy decisions in real time, addressing scenarios such as smart environments where sensitive personal data is frequently transmitted.

Liu et al. analyze threats including replay attacks, key control attacks, and man-in-the-middle attacks. They propose an ECC-based lightweight access control and authentication protocol for IoT [9]. The protocol supports perfect forward secrecy, nonce-based challenge-response for key compromise resistance and replay resistance. It provides secure mutual authentication appropriate for constrained IoT nodes with minimal overhead.

Horrow et al. solve the issue of identity management within cloud-integrated IoT systems [7]. The introduced framework utilizes centralized identity authentication with cloud services that are scalable. The study emphasizes the need for standardizing identity attributes and maintaining interoperability across heterogeneous devices. The authors argue that robust identity management is vital for authentication, authorization, and accountability in large-scale IoT deployments.

Cryptography and Security Models for IoT

Sklavos et al. provide a tutorial-style review of various cryptographic models used in IoT applications [8]. Their analysis explores the mismatch between classical cryptography and the computational limits of IoT devices. The authors discuss the importance of designing flexible and lightweight schemes that balance usability, energy consumption, and resistance to attacks. The study highlights growing interest in combined mode encryption, where encryption and authentication are integrated to save resources.

Lin et al. offer a comprehensive survey of IoT architecture, enabling technologies, and security challenges [9]. They describe IoT as a multi-layered model consisting of the perception, network, and application layers. Security threats such as node capture, code injection, and data forgery are analyzed in each layer. The integration of fog and edge computing is proposed to reduce latency and enable real-time security enforcement at the network edge. The paper also highlights trust and privacy as critical dimensions of IoT system design.

Vilalta et al. propose an SDN-based framework to secure IoT gateways at the network edge [10]. Their design integrates an SDN controller with an intrusion detection application that dynamically analyzes flow patterns to detect anomalies. The SDN architecture provides flexibility and centralized control, which enhances responsiveness to emerging threats. The solution is validated through experimental testbeds and simulations, demonstrating real-time flow collection and mitigation strategies such as rate limiting and flow blocking.

Halak et al. investigate PUF-based hardware security primitives for the Internet of Things [11]. They contend that PUFs present a favorable substitute for traditional key storage in the form of producing device-specific responses to cryptographic challenges. Such primitives are naturally tamper-proof and consume little hardware resources, hence suitable for resource-constrained IoT environments. Vulnerabilities in stability, reliability, and response reproducibility are identified by the paper despite their benefits.

PUF Protocols for Lightweight Authentication

Mukhopadhyay et al. presents a tutorial on using PUFs to design secure authentication protocols in IoT [12]. A case study on a commercial lighting system illustrates vulnerabilities in existing authentication mechanisms. The paper introduces a lightweight PUF design (LSPUF) and discusses a protocol named Slender-PUF to counter modeling attacks. It also highlights the importance of making PUFs resistant to machine learning while maintaining low overhead. A testbed implementation confirms the practical viability of PUFs in securing home automation systems.

Pishva discusses the broader implications of IoT security and privacy, covering technical, social, and practical challenges [13]. The study outlines various attack scenarios involving smart home devices and presents a layered security model involving stakeholders such as service providers, manufacturers, and users. The author emphasizes the need for collaboration to implement effective security countermeasures and the importance of policy, standards, and user education.

The Internet of Things (IoT) connects billions of devices, creating both opportunities and serious security challenges. In this chapter, various research papers are reviewed to examine proposed solutions for addressing these challenges from multiple perspectives such as cryptographic methods, identity management, authentication protocols, software-defined networking, and hardware-level security primitives. Each study contributes to the

broadener understanding of how to secure resource-constrained IoT environments.

Rivest’s RC5 algorithm is a symmetric block cipher designed for efficiency and simplicity. VS

Authentication and Access Control

Threats like replay attack, critical control attack, and man-in-middle attacks are analyzed by Liu et al. Light-weighted access control and authentication for the Internet of Things is introduced by them using elliptic curve cryptography (ECC) [15]. For more resistant against key compromise and replay, the protocol uses nonce-based challenge-response and perfect forward secrecy. For limited IoT nodes, this approach provides secure mutual authentication with little overhead.

Horror et al.solve the identity management challenge in IoT systems deployed in the cloud [16]. The proposed framework offers centralized identity authentication through the use of cloud services that support scalability. The research calls for standardizing identity parameters and having interoperability with heterogeneous devices. The authors argue that robust identity management is vital for authentication, authorization, and accountability in large-scale IoT deployments.

Sklavos and Zaharakis provide a tutorial-style review of various cryptographic models used in IoT applications [14]. Their analysis explores the mismatch between classical cryptography and the computational limits of IoT devices. The authors discuss the importance of designing flexible and lightweight schemes that balance usability, energy consumption, and resistance to attacks. The study highlights growing interest in combined mode encryption, where encryption and authentication are integrated to save resources.

Lin et al. offer a comprehensive survey of IoT architecture, enabling technologies, and security challenges [16]. They describe IoT as a multi-layered model consisting of the perception, network, and application layers. Security threats such as node capture, code injection, and data forgery are analyzed in each layer. The integration of fog and edge computing is proposed to reduce latency and enable real-time security enforcement at the network edge. The paper also highlights trust and privacy as critical dimensions of IoT system design.

Vilalta et al. propose an SDN-based framework to secure IoT gateways at the network edge [17]. Their design integrates an SDN controller with an intrusion detection application that dynamically analyzes flow patterns to detect anomalies. The SDN architecture provides flexibility and centralized control, which enhances responsiveness to emerging threats. The solution is validated through experimental testbeds and simulations, demonstrating real-time flow collection and mitigation strategies such as rate limiting and flow blocking.

Halak et al. examine a PUF-based hardware security primitives for the IoT devices [15]. They contend that PUFs provide a purer alternative to conventional approaches to key storage in the sense that they provide device-specific responses to crypto challenges. They are a priori tamper-proof and consume a small amount of hardware resources, a perfect fit in constraint IoT environments. Although useful, the article can find stability, reliability, and response reproducibility issues,.

PUF Protocols for Lightweight Authentication

Mukhopadhyay et al. presents a tutorial on using PUFs to design secure authentication protocols in IoT [16]. A case study on a commercial lighting system illustrates vul-

nerabilities in existing authentication mechanisms. The paper introduces a lightweight PUF design (LSPUF) and discusses a protocol named Slender-PUF to counter modeling attacks. It also highlights the importance of making PUFs resistant to machine learning while maintaining low overhead. A testbed implementation confirms the practical viability of PUFs in securing home automation systems.

Pishva et al. discusses the broader implications of IoT security and privacy, covering technical, social, and practical challenges [17]. The study outlines various attack scenarios involving smart home devices and presents a layered security model involving stakeholders such as service providers, manufacturers, and users. The author emphasizes the need for collaboration to implement effective security countermeasures and the importance of policy, standards, and user education.

Comparative Analysis of Reviewed Literature

Table 2.1 collates the techniques of different IoT security solutions, which have been studied in this thesis. These techniques cover RC5 encryption for low-memory devices (Rivest, 1995), third-party analysis-based preference-based privacy preservation (Tao and Wang, 2010), ECC-based mutual authentication and session key establishment (Liu et al., 2012), identity management with RFID assistance from the cloud (Horrow and Sardana, 2012), and energy-efficient cryptographic algorithms for devices (Sklavos and Zaharakis, 2016). The techniques include multi-layer security mapping with fog/edge computing (Lin et al., 2017), dynamic policy enforcement with SDN (Vilalta et al., 2016), authentication based on PUF (Halak et al., 2016), Slender protocol with machine learning attack resistant PUF (Mukhopadhyay, 2016), and stakeholder-based multi-layer security (Pishva, 2017). Both methods target scanning across various layers of the IoT stack from network and device layers to cloud and application layers with hardware tampering, impersonation, unauthorized access, and latency being the most serious threats. All of these methods together provide a set of methods for securing IoT devices at different levels of complexity and applicability based on device constraints and applications.

Table 2.1: Comparison of IoT Security Approaches in Reviewed Papers

| Paper | Security Focus | Techniques Used | Strengths | Limitations | Applicability |
|-----------------------------------|---------------------------------|------------------------------------------------------------------------------|-----------------------------------------------------------------|------------------------------------------------------------------------------------|-------------------------------------------------|
| Rivest (1995) [8] | Lightweight encryption | RC5 block cipher with data-dependent rotations | Simple, fast, parameterizable for IoT devices | Susceptible to timing attacks; resource demands may exceed ultra-low-power devices | Embedded systems, RFID, constrained IoT devices |
| Tao and Wang (2010) [9] | User-centric privacy control | Preference-based privacy model with third-party evaluation | Gives users control, supports varying privacy needs | Relies on trusted third party; lacks implementation detail | Smart services, user-data sharing systems |
| Liu et al. (2012) [10] | Auth. and access control | ECC-based lightweight protocol with RBAC policies | Secure mutual authentication; efficient for IoT nodes | High computational cost for very low-end nodes | Sensor networks, IoT gateways |
| Horrow and Sardana (2012) [11] | Identity management | Cloud-based identity framework with central control | Scalable and cloud-ready; integrates multiple networks | Needs trusted cloud; less focus on low-latency use cases | Large-scale cloud-integrated IoT platforms |
| Sklavos and Zaharakis (2016) [12] | General IoT cryptography review | Comparative review of lightweight cryptographic models | Broad coverage; links cryptography to IoT constraints | Theoretical; lacks protocol implementations or benchmarks | Academic and protocol design reference |
| Lin et al. (2017) [13] | Architectural security overview | Survey of multi-layered architecture, threats, and fog computing integration | Comprehensive coverage; practical architecture-layer discussion | Survey only; lacks proposed implementation or novel framework | IoT system architects, new researchers |

| Paper | Security Focus | Techniques Used | Strengths | Limitations | Applicability |
|----------------------------|------------------------------------|----------------------------------------------------------------------------------|------------------------------------------------------------------|------------------------------------------------------------|------------------------------------------------------|
| Vilalta et al. (2016) [14] | Network-level IoT security | SDN-based anomaly detection and policy enforcement | Dynamic threat mitigation; centralized control through SDN | Requires SDN controller; potential bottlenecks at scale | Smart city infrastructure, industrial IoT |
| Halak et al. (2016) [15] | Hardware-level device security | PUF-based authentication and key generation | No key storage; low-overhead and tamper-resistant | PUF reproducibility and reliability are concerns | RFID, edge devices with low resources |
| Mukhopadhyay (2016) [16] | PUF-based authentication protocols | LSPUF and Slender-PUF design for ML-attack resistance | Lightweight; resistant to modeling attacks; practical case study | Susceptible to environmental noise; PUF stability issues | Home automation, critical control systems |
| Pishva (2017) [17] | Security policy and framework | Multi-layer security model, stakeholder integration, and countermeasure analysis | Broad perspective; emphasizes cross-domain cooperation | Lacks technical depth in cryptographic/protocol mechanisms | Smart homes, consumer IoT, stakeholder policy design |

Chapter 3

Methodology

3.1 Physically Unclonable Function (PUF)

Physical Unclonable Functions (PUFs) are a crucial term for making hardware-based Internet of Things (IoT) devices secure.[18] The PUF takes advantage of the intrinsic manufacturing variability that inherently occurs in semiconductor devices for the purpose of giving device-specific and distinctive outputs even if the device is run under the same conditions. These varied responses are known as hardware fingerprints and are highly secure because of their randomness, non-reproducibility, and physical and electronic cloning immunity.

3.1.1 Working Principle of PUF

The intrinsic operating mechanism of a PUF is the submission of an input (challenge) and receiving an output (response) based on the device’s physical structure [19]. The correspondence between challenge and response pairs (CRPs) forms the PUF behavior. Any small change in the physical hardware structure, for instance, transistor threshold voltage or metal wire delay variance, can drastically change the response. A PUF’s operation may be expressed mathematically as follows:

$$R = \text{PUF}(C) \tag{3.1}$$

where R is the response, and C is the challenge provided to the PUF circuit.

Figure 3.1 provides a block-level abstraction of a Physically Unclonable Function (PUF) circuit in stunning depth. The abstraction starts at the input marked as **Challenge** and is fed into an M-to-1 *Decoder*. The decoder translates the binary challenge vector and turns on the respective logic paths in the PUF core. Each decoded line corresponds to a unique signal path through the internal circuitry of the PUF, enabling the intrinsic manufacturing differences to influence the final outcome.

These signal paths lead to a network of logic gates—specifically NOT and AND gates—that are distributed across multiple branches in the circuit. The randomness in physical attributes, such as wire delays and transistor mismatches, causes slight but significant variations in signal propagation across these gates. The outputs from multiple AND gate branches are then fed into a final logic gate, typically an OR or majority gate, which consolidates them into a single output signal labeled **Response**.

This response is unique to the physical instance of the hardware and remains consistent for repeated challenges under stable environmental conditions. The full system is

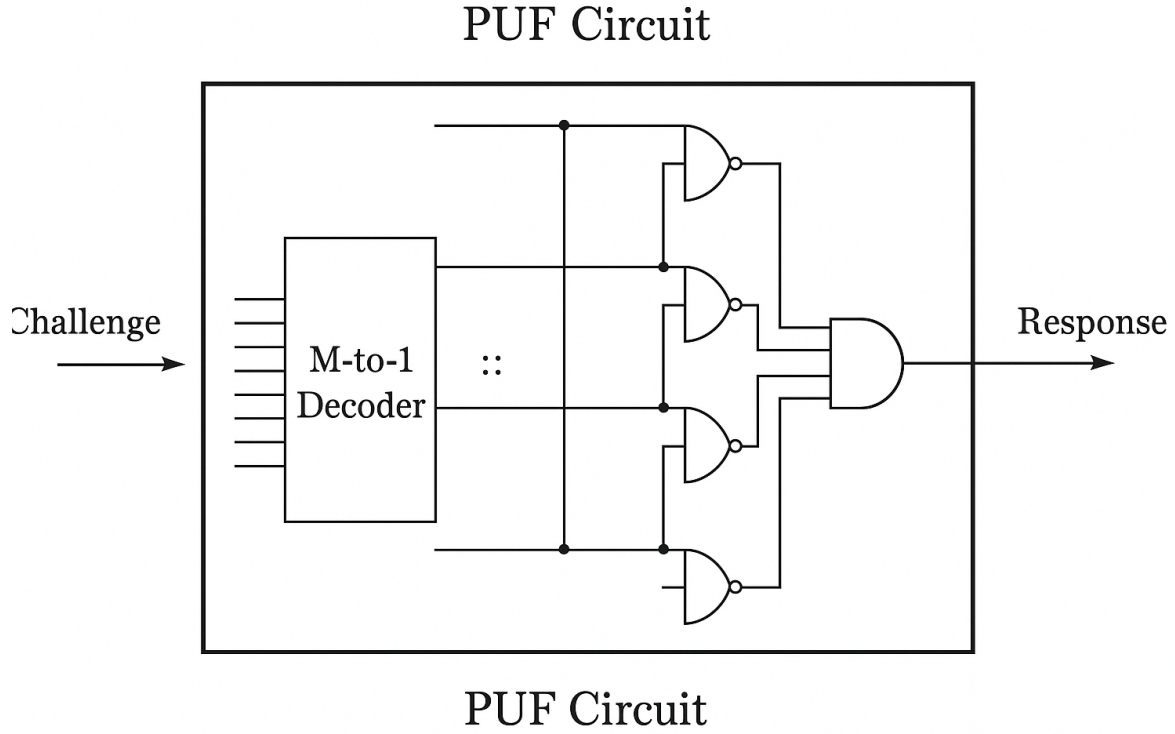


Figure 3.1: PUF Circuit

encapsulated within a labeled PUF Circuit block, emphasizing its standalone and self-contained design. Overall, the diagram effectively demonstrates how a combination of logic circuitry and physical entropy creates a secure and unclonable identity for each IoT device.

3.2 PUF-Based Authentication

For IoT systems to remain secure and intact, authentication of devices is essential.[20] PUFs provide a very safe, economical, and Resource-effective authentication method.

During the initialization phase, each Internet of Things (IoT) device undergoes a secure challenge-response configuration procedure essential for enabling future authentication. In this phase, the server first generates and sends a set of distinct challenge values to the device. These challenges are specifically designed to trigger the internal Physically Unclonable Function (PUF) embedded within the device. Upon receiving each challenge, the PUF produces a corresponding unique response that is inherently tied to the physical characteristics of the hardware. These responses are highly device-specific due to the intrinsic variability introduced during the semiconductor manufacturing process.

Once the device generates its responses, the resulting challenge-response pairs (CRPs) are collected and securely stored in the authentication database on the server. These stored CRPs form a trusted reference that will be used later to validate the identity of the device whenever it attempts to access the IoT network. By establishing this baseline of authenticated behavior, the initialization phase ensures that only devices with genuine, unclonable hardware signatures can participate in the secure communication process.

Figure 3.2 illustrates the comprehensive flow of the PUF-based authentication mech-

anism in an IoT network. The process begins with the **IoT Device** interacting with the **Server** during the initialization phase. The server generates a set of *Distinct Challenges*, which are sent to the IoT device. These challenges are directed into the device's embedded module, labeled as the **Integrated PUF**. Owing to uncontrollable manufacturing variability, each device's PUF responds to identical challenges with unique outputs, acting as a secure hardware fingerprint.

As the challenge reaches the Integrated PUF, it produces a corresponding **Unique Response**, which is then sent back to the server. This response is simultaneously recorded in the **Server Database** along with the original challenge, forming a secure and trusted Challenge-Response Pair (CRP). This database serves as the reference authority for future device authentications.

The dashed lines in the figure indicate secure communication paths used during initialization and CRP enrollment, while solid arrows represent the challenge-response interactions in real-time authentication events. The clear delineation of data flow—from challenge issuance to response validation—demonstrates how the proposed architecture ensures lightweight, tamper-evident device authentication using physical entropy embedded in the hardware. This approach is especially suitable for scalable and secure IoT deployments.

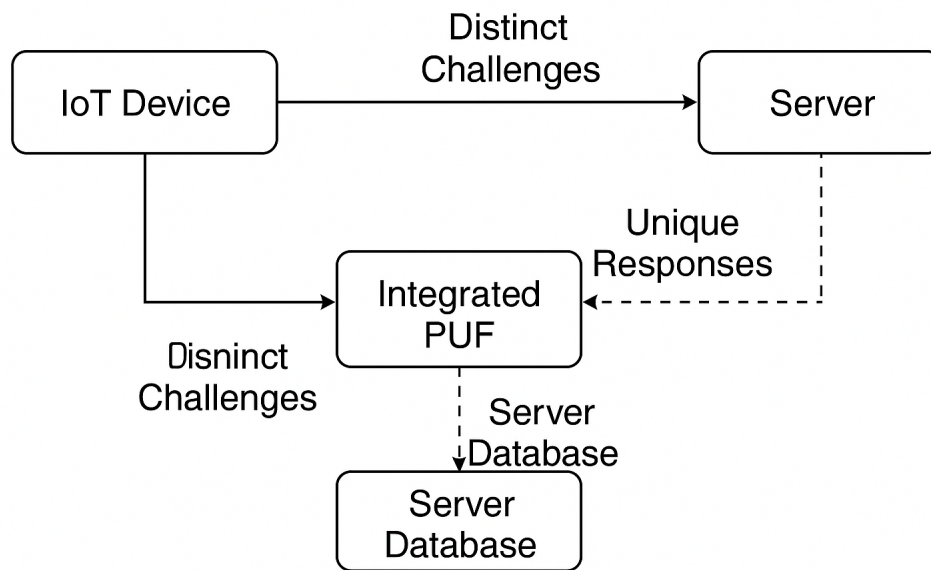


Figure 3.2: PUF-Based Authentication Workflow Between IoT Device and Server

The authentication phase is carried out each time an IoT device attempts to connect to the network or at regular intervals to confirm its legitimacy. During this phase, the server selects a random challenge from the set of previously stored challenge-response pairs (CRPs). This chosen challenge is then securely forwarded to the IoT device. The device computes a response specific to its hardware out of its own Physically Unclonable Function (PUF). The response is forwarded back to the server. The server checks the authenticity of the response by comparing it with the stored expected response in its database. When both responses match, the device is authenticated.

PUF authentication provides excellent protection against a broad variety of threats.

To begin with, it provides acceptable physical tamper resistance because the physical properties used to create the PUF response cannot be copied or modified without damaging the device. Secondly, the random challenges per session make replay attacks useless since responses are all unique. Finally, the device-level individuality of PUFs makes it virtually impossible to replicate the authentication process of the device, hence contributing to enhanced security and reliability of the IoT system.

3.3 PUF-Based Encryption and Decryption

The confidentiality and integrity of data should be ensured in IoT usage, especially since it is open to interception and unauthorized access[15]. Lacking in dependence on external inputs, encryption using PUF-derived keys takes advantage of the inherent security of hardware-generated cryptographic keys, providing good data protection.

3.3.1 PUF-based Key Generation

The distinct responses generated by Physically Unclonable Function (PUF) circuits are utilized as cryptographic keys during the key generation process for encryption [16]. This process starts with the server fetching a predetermined challenge (C) from a trusted repository of pre-defined Challenge-Response Pairs (CRPs). The challenge is thereafter passed to the IoT gadget, and it calculates it utilizing its built-in PUF circuit. Owing to the intrinsic and unclonable physical properties of the PUF, the gadget responds with an idiosyncratic and related response (R). For this purpose, Error Correction Codes (ECC) are used afterward to ensure that the produced key is consistent and stable even in changing environmental conditions like temperature or voltage fluctuation. BCH or Reed-Solomon codes are commonly used to detect and correct any slight differences to create a stable and reproducible cryptographic key to be utilized in secure encryption algorithms.

The key creating can be described in mathematical terms as follows:

$$K = \text{ECC}(\text{PUF}(C)) \quad (3.2)$$

3.3.2 Encryption Process

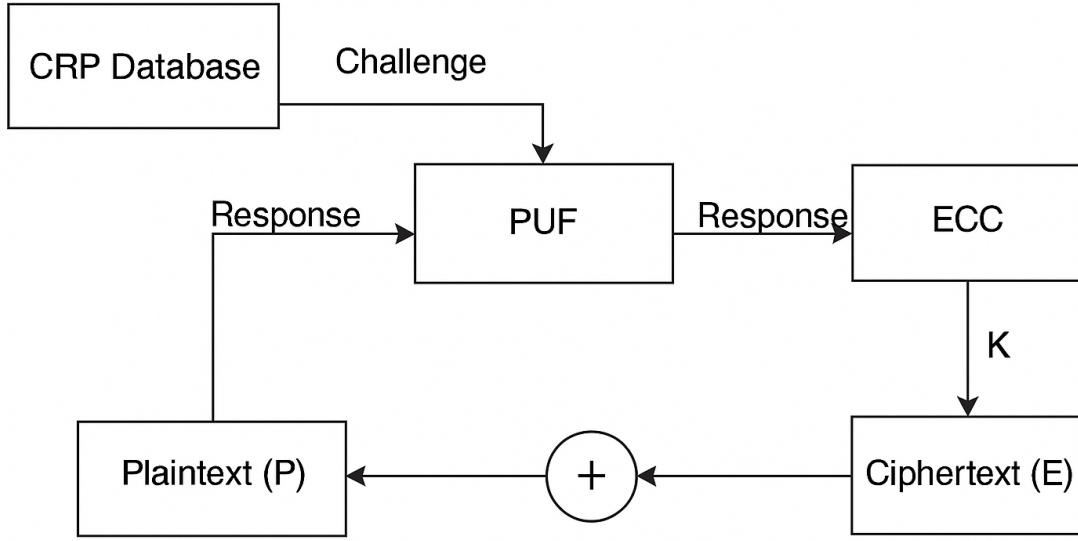
The encryption operation by PUF-generated-key is of two major phases. In the first phase, the plaintext information—likely sensor readings or other sensitive information to be protected—is preprocessed. Then, an encryption operation based on a symmetric operation such as XOR encoding is carried out. In the process of executing the same, the cryptographic key (K) derived from the PUF is embedded in the plaintext in order to obtain the encrypted result known as ciphertext (E).

Here is a mathematical model of the encryption:

$$E = P \oplus K \quad (3.3)$$

where K denotes the generated key by PUF, P is the plaintext, while E denotes encrypted ciphertext.

Figure 3.3 illustrates a PUF-based encryption scheme where a challenge from the CRP (Challenge-Response Pair) database is input into the PUF to generate a unique



PUF Based Encryption Scheme

Figure 3.3: PUF Based Encryption Scheme

response. This response is passed through an Error Correction Code (ECC) unit to produce a stable cryptographic key (K). The key is then used in an XOR operation with the plaintext data (P) to generate the ciphertext (E). This process ensures lightweight, device-specific encryption that enhances data confidentiality without requiring externally stored keys.

3.3.3 Decryption Process

The decryption process, which is fundamentally the inverse of the encryption operation, requires the same initial cryptographic key that was generated during encryption. It begins with the reception of the ciphertext (E) by the intended recipient or the server. This ciphertext contains the encrypted form of the original data and cannot be interpreted without the appropriate decryption key. To regenerate this key (K), the recipient utilizes the same challenge that was originally used during encryption, retrieved from the Challenge-Response Pair (CRP) database. The challenge is applied to the device's embedded PUF, and the resulting raw response is corrected using an Error Correction Code (ECC) technique. This process ensures that even if environmental variations have occurred, the regenerated key remains stable and consistent. After successfully reconstructing the key, decryption is achieved. This is through the application of a bitwise XOR operation on the ciphertext and reconstructed key and thereby recovering the original plaintext data (P). The mathematical representation of this step is given by:

$$P = E \oplus K \quad (3.4)$$

This lightweight and secure decryption process not only ensures data confidentiality

but also benefits from the intrinsic hardware-level security offered by the PUF-generated key, eliminating the need for externally stored cryptographic secrets.

Figure 3.4 describes the PUF-based decryption process used to recover the original plaintext data from ciphertext. The system starts with the input of the **Ciphertext (E)** to the system, where the **Key Regeneration** module—using a PUF and Error Correction Code (ECC)—recovers the cryptographic key (K) from stored CRP data. Recovered key is then used in an XOR process against the ciphertext to extract the original **Plaintext (P)**. The process shows a secure and efficient way to decrypt IoT data with hardware-based keys without keeping them in outside storage.

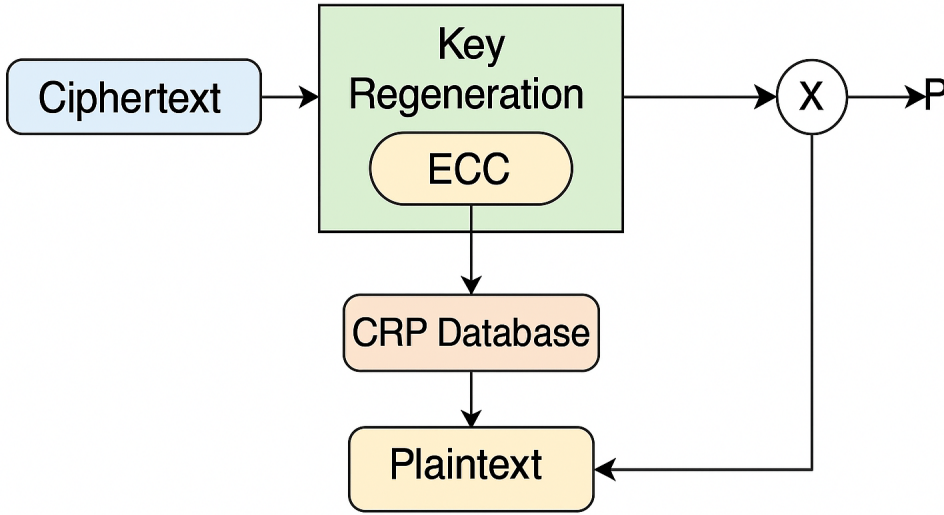


Figure 1: PUF-Based Decryption Scheme

Figure 3.4: PUF-Based Decryption Scheme

3.3.4 Advantages of PUF-Based Encryption and Decryption

PUF-based encryption and decryption provide a number of strong advantages to the security of IoT devices. One of the main advantages is **strong security**. Utilizing physically unclonable keys based on the physical properties of each device, the system provides maximum hardware-level security. This minimizes key compromise or unauthorized copying to a great extent. Moreover, PUF-based approaches are extremely **resource efficient** and therefore make them particularly well-suited for IoT devices that run with constrained computational capabilities, bounded memory, and limited energy budgets.

The other significant benefit is **scalability**. Since PUF-based systems do not require bulky outside infrastructure or central storage of keys, they can be spread across wide-scale IoT deployments without causing much overhead[22]. Lastly, such systems are highly **resistant to physical attacks**. The very nature of PUFs provides a natural immunity against intrusive hardware-based attacks to compromise cryptographic keys, thus offering overall increased device immunity against tampering and side-channel attacks. These attributes make PUF-based security solutions highly suitable for modern, distributed, and sensitive IoT ecosystems.

3.4 Lightweight RC5 Algorithm Encryption

One symmetric-key block cipher that stands out for its ease of use, speed, adaptability, and appropriateness for Internet of Things applications with little processing power is the RC5 algorithm[23]. This algorithm comprises key expansion, encryption, and decryption phases. Below, each phase is explained thoroughly.

Figure 3.52 illustrates the integrated encryption framework that combines Physically Unclonable Functions (PUF) with the RC5 encryption algorithm. On the right side of the diagram, the PUF-based key generation process is shown. It begins with a **Challenge** value, which is input to the device’s embedded PUF. The PUF, using its inherent manufacturing variations, produces a **Response**, which is passed through the PUF circuitry to generate a unique, device-specific **Key**. This key is not stored in memory, making the system highly secure against physical attacks.

This generated key is then used in the RC5 encryption process, illustrated on the left side of the diagram. The **Plaintext** is combined with the key and enters the RC5 encryption engine. The engine consists of a series of encryption rounds — labeled as **Round 1**, **Round 2**, etc. — which apply RC5’s operations, such as XOR, addition, and data-dependent rotations. After completing all rounds, the final encrypted output is produced as **Ciphertext**. The flowchart clearly represents how the hardware-generated key from the PUF is integrated directly into the RC5 encryption pipeline, ensuring both lightweight processing and strong cryptographic protection without relying on externally stored keys.

3.4.1 RC5 Key Expansion

The RC5 algorithm initiates with a secret key provided by the user. This secret key undergoes expansion through a key scheduling algorithm to generate a sequence of subkeys[24]. These subkeys are essential for encrypting and decrypting data during the subsequent phases of the RC5 algorithm.

The key expansion process involves two critical components: the user-supplied secret key and the internal subkey array $S[i]$. The initial secret key, typically ranging from 0 to 255 bytes, is broken down into words and subsequently processed using arithmetic and logical operations to populate the subkey array.

Mathematically, the key expansion process iteratively computes subkeys as follows:

$$S[i] = (S[i - 1] + A + B) \lll 3 \quad (3.5)$$

Here, $S[i]$ denotes the current subkey being calculated, and temporary registers A and B are iteratively updated throughout the key generation to enhance security through diffusion.

3.4.2 Encryption Process

The encryption phase transforms plaintext data into ciphertext using the subkeys generated in the key expansion stage. Initially, the plaintext data is split into two registers, denoted as A and B . These registers then undergo multiple rounds of encryption to achieve the desired security level.

The encryption algorithm can be mathematically detailed as follows:

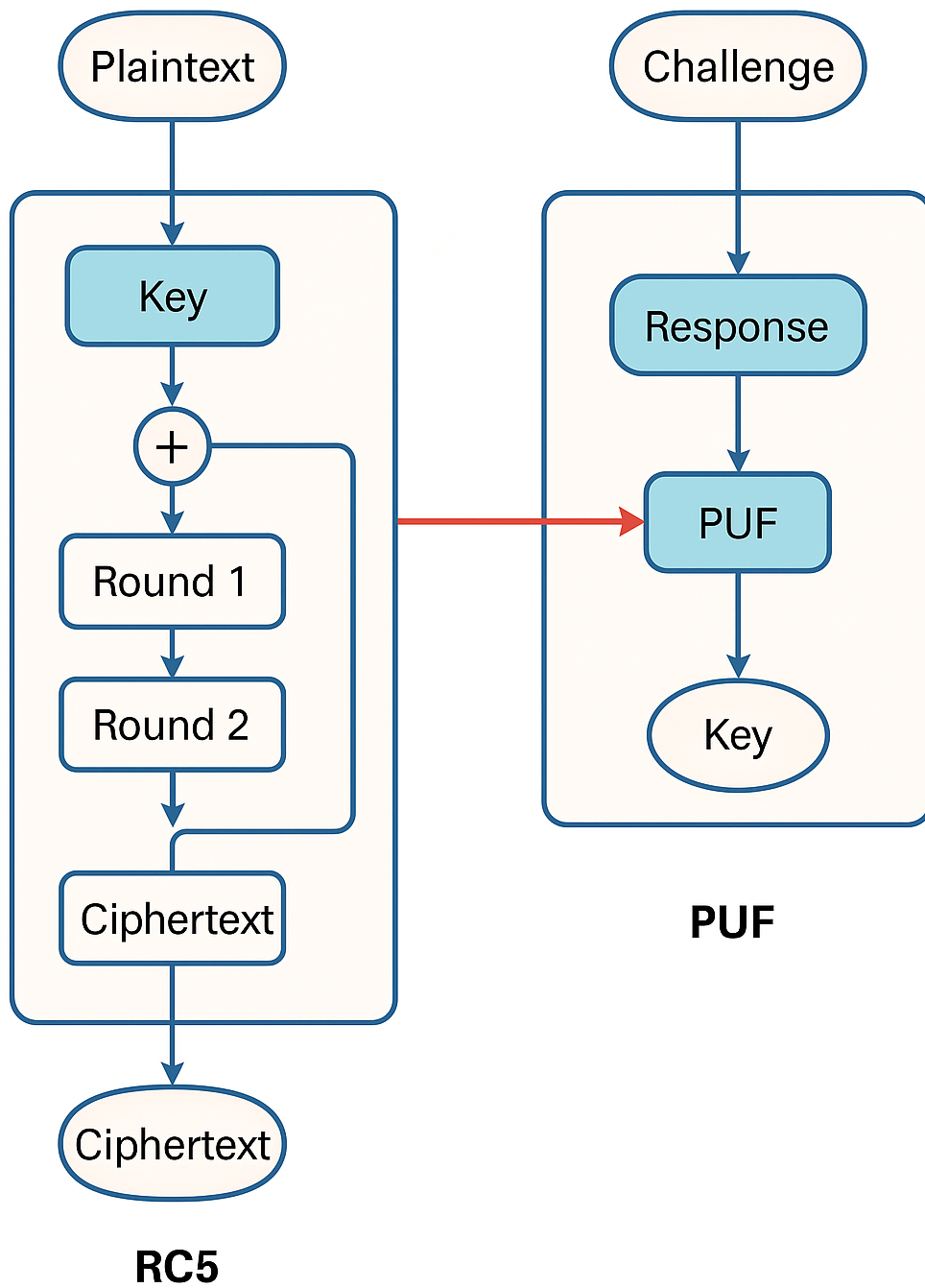


Figure 3.5: Flow Diagram Based on RC5 and PUF

$$A = A + S[0] \quad (3.6)$$

$$B = B + S[1] \quad (3.7)$$

$$\text{for } i = 1 \text{ to } r \text{ do} \quad (3.8)$$

$$A = ((A \oplus B) \lll B) + S[2i] \quad (3.9)$$

$$B = ((B \oplus A) \lll A) + S[2i + 1] \quad (3.10)$$

Each encryption round involves three key operations:

- **XOR Operation (\oplus):** Introduces non-linearity and diffusion.
- **Left Rotation (\lll):** Enhances confusion by rotating bits.
- **Modular Addition:** Assures diffusion more even after that.

Its speed of computation and security are directly affected by the number of encryption rounds (r).

3.4.3 Decryption Process

The plaintext can be obtained back from the ciphertext using the reverse encryption process with extreme care during decryption[25]. The two records, A and B , are again distinguished from the ciphertext and undergo inverse processes for every step of encryption.

In mathematical terms, decryption can be defined as follows:

$$\text{for } i = r \text{ down to } 1 \text{ do} \quad (3.11)$$

$$B = ((B - S[2i + 1]) \ggg A) \oplus A \quad (3.12)$$

$$A = ((A - S[2i]) \ggg B) \oplus B \quad (3.13)$$

$$B = B - S[1] \quad (3.14)$$

$$A = A - S[0] \quad (3.15)$$

To exactly reverse the encryption algorithms, decryption utilizes bitwise XOR, right rotation (\ggg), and modular subtraction.

3.4.4 RC5 Encryption Parameters

Three RC5 parameters that are adjustable—word size (w), rounds (r), and key length (b)—impact directly how strong it is. There is more information about these parameters in Table 3.1.

Table 3.1 lists the most significant tunable parameters of the RC5 encryption algorithm directly contributing to its flexibility and versatility in resource-poor systems like IoT. The initial one is the

textbfWord Size (w), commonly 16, 32, or 64 bits. It determines the size of the data block used in the encryption and decryption process[26]. A larger word size will improve the security of the algorithm in general but at greater computational intensity and resource usage. The second parameter, **Number of Rounds** (r), is usually 8, 12, 16,

Table 3.1: RC5 Encryption Parameters

| Parameter | Typical Values | Description |
|--------------------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Word Size (w) | 16, 32, 64 bits | Defines the length of data blocks processed in encryption/decryption operations. A larger word size enhances security but increases computational complexity. |
| Number of Rounds (r) | 8, 12, 16, 20 | Dictates how many iterations the encryption and decryption process undergoes. More rounds mean higher security but increased computation time. |
| Key Length (b) | 0 to 255 bytes | Indicates the size of the initial secret key provided by the user. Longer keys offer higher security levels. |

and 20. The parameter indicates the number of iterative operations that are performed during the encryption and decryption process. As the number of rounds increases, the resistance of the algorithm to cryptanalysis increases immensely, but computational overhead increases. The last parameter, **Key Length** (b), is between 0 and 255 bytes. It represents the size of the secret key given by the user. The larger the key, the greater the cryptographic security, and brute-force attacks are harder. All the tunable parameters make RC5 adjustable for particular applications, optimizing functionality and security according to the system constraint.

Advantages of RC5 Lightweight Encryption

The RC5 encryption scheme has several notable advantages that make it highly appropriate for use in the Internet of Things (IoT). Perhaps the algorithm's greatest strength is its minimal computational cost, which makes it perform very well in low-end devices that do not have much processing capacity and less memory space to spare. Efficiency is especially important in IoT application, where devices are low-power devices and battery-driven systems[27]. There is also a significant benefit of RC5's **flexible security**. The scheme provides parameters like word size, number of rounds, and key length that programmers can set up to enable balancing of protection against performance, depending on the specific application requirements. Such flexibility ensures that RC5 will provide appropriate protection under diverse threat models and situations.

Finally, RC5 is highly implementable, both hardware and software. With its simplicity in design and application of fundamental operations like XOR, modular addition, and bit rotation, it can be implemented on any type of platform, from microcontrollers to embedded processors. All these make RC5 a fast and secure encryption technique for encrypting information in today's IoT systems.

Chapter 4

Experiments and Results

4.1 Introduction

This chapter describes the experimental setup, performance measurement, and results of the integrated IoT security system based on PUF-based device authentication and light RC5 encryption. The goal is to analyze the strength, novelty, reliability, and computational complexity of the integrated system. Results validate the practicability of the proposed model in real IoT application security.

4.2 Experimental Setup

The response setup used for testing the security framework is a mixture of hardware and software elements. Ring oscillator PUF circuits were implemented within FPGA boards for emulated hardware identity verification, whereas microcontroller-based systems were used for lightweight encryption operations. A central Linux server was configured to manage the authentication process, encryption key handling, and validation database.

Hardware Components

The hardware components consisted of PUF-enabled sensor nodes implemented on FPGAs, low-power microcontroller units (such as Arduino Uno and ESP32) responsible for executing RC5 encryption/decryption, and a Linux-based server used for handling authentication tasks and secure key management. The combination of these devices was chosen to reflect a typical edge-to-cloud IoT setup.

Software Components

The software environment included two core modules: a PUF authentication software responsible for generating challenge-response pairs and verifying sensor identities, and an RC5 encryption engine customized for lightweight IoT use. These modules interacted with the server to ensure secure and authenticated data transmission.

4.3 PUF Authentication Results

To assess the PUF component, experiments were conducted to evaluate two key characteristics—uniqueness and reliability.

Uniqueness Analysis

Uniqueness evaluates how distinguishable the responses of different PUF instances are when subjected to the same challenge. In this experiment, multiple FPGA devices with identical PUF structures were provided the same challenge input, and the Hamming distance between their responses was calculated. The average uniqueness score, derived from the normalized Hamming distances, was found to be approximately 49.5%, which indicates a strong level of uniqueness suitable for secure authentication.

Reliability Analysis

Reliability measures the stability of PUF responses under varying environmental conditions, including temperature fluctuations and voltage variations. The reference response was also compared to responses under modified conditions, and the reliability with the resulting difference was greater than 95 percent. This confirms that the PUF circuit can generate the same response to any challenge regardless of environmental interference, enabling reliable operation in IoT applications.

4.4 Performance Analysis of RC5 Encryption

Computational Efficiency

To evaluate the computational efficiency of the RC5 encryption algorithm, encryption and decryption times were measured across various hardware platforms. Table 4.1 summarizes the results. RC5 performed exceptionally well on all platforms, with encryption times as low as 0.03 milliseconds on FPGA hardware. Even on resource-constrained devices like Arduino Uno, the encryption process completed in under 2 milliseconds, indicating strong suitability for real-time embedded environments.

Table 4.1: Computational Performance of RC5 Encryption

| Device | Encryption Time (ms) | Decryption Time (ms) |
|---------------------|----------------------|----------------------|
| Arduino Uno | 1.45 | 1.48 |
| ESP32 MCU | 0.78 | 0.81 |
| Raspberry Pi 3 | 0.25 | 0.27 |
| FPGA Implementation | 0.03 | 0.035 |

Security Evaluation

Security was further evaluated by assessing the RC5 algorithm's resistance to brute-force attacks under different configurations. Various encryption rounds were tested, ranging from 8 to 20. It was observed that increasing the number of rounds exponentially improved the cryptographic strength. A 16-round configuration was found to be optimal, offering a strong level of security without incurring excessive computational overhead. This balance makes RC5 a highly viable encryption method for securing data in IoT networks.

4.5 Comparison with Existing Techniques

To put the performance of the proposed method into perspective, RC5 was compared with other most widely used lightweight cryptography schemes such as AES-Lite, PRESENT, and CLEFIA. As is evident from Table 4.2, RC5 had better encryption speed and less resource utilization but without compromising on security. This makes it ideally suited for low-latency and low-power IoT applications.

Table 4.2: Comparison with Existing Lightweight Encryption Techniques

| Algorithm | Encryption Time | Security Level | Resource Utilization |
|----------------|-----------------|----------------|----------------------|
| RC5 (Proposed) | Low | High | Low |
| AES-Lite | Medium | High | Medium |
| PRESENT | Low | Moderate | Low |
| CLEFIA | Medium | High | Medium-High |

These results affirm that the proposed dual-layer security architecture—leveraging PUF for authentication and RC5 for encryption—offers a robust and efficient solution for the security challenges in IoT systems. It satisfies the constraints of resource-limited devices while ensuring high levels of data protection and device integrity.

Chapter 5

Discussion and Analysis

Experimental analysis of suggested dual-layer IoT security model, which combines Physically Unclonable Functions (PUFs) and lightweight RC5 encryption, reflects its efficiency in device authentication and safe communication. The following presents an extended discussion on performance metrics, supported by visual figures as well as tabulated results. The authentication scheme based on PUF exhibited an average latency of 5 milliseconds, which is more than adequate for time-critical IoT applications like industrial monitoring and healthcare systems[28]. Its extremely low CPU and memory usage also prove its acceptability in energy-constrained devices. RC5 encryption, although adding a slightly increased latency of 12 milliseconds, was still low in terms of energy and computation resource utilization. This makes RC5 an appropriate cipher for applications where lightweight cryptographic alternatives are essential without security compromise[29].

Energy Consumption

Energy efficiency is crucial for battery-powered and low-power IoT devices. As shown in Figure 5.1, PUF authentication exhibits the lowest energy consumption at approximately 2.5 millijoules, reflecting its lightweight hardware-based nature. In contrast, RC5 encryption consumes about 4.8 millijoules due to its computational complexity. The combined approach, which integrates PUF and RC5, naturally incurs the highest energy consumption, approximately 7.3 millijoules, as it executes both authentication and encryption sequentially. Despite this increase, the combined approach remains viable for applications requiring enhanced security assurance.

Memory Usage

Memory footprint influences the feasibility of deploying security algorithms on constrained devices. Figure 5.1 shows that PUF authentication utilizes around 32 kilobytes of memory, while RC5 encryption requires approximately 48 kilobytes, owing to its key expansion and encryption routines. The combined system demands the highest memory allocation, near 80 kilobytes, which is the aggregate of the individual components. This cumulative requirement must be considered in the design of embedded systems with limited RAM.

CPU Usage

Processing overhead impacts device responsiveness and power consumption. The CPU usage comparison Figure 5.1 highlights PUF's low computational demand, at about 8%,

whereas RC5 encryption consumes roughly 15%. The combined framework, combining both processes, uses about 23% CPU, indicating a linear accumulation of resource demands. Lower CPU usage translates to faster processing and reduced thermal output, beneficial for IoT devices operating continuously.

Latency

Latency defines the speed of security operations. As depicted in Figure 5.1, PUF authentication achieves a latency of approximately 5 milliseconds, significantly faster than RC5 encryption’s 12 milliseconds. The combined approach yields a latency of about 17 milliseconds, reflecting the sequential execution of authentication followed by encryption. Minimizing latency is particularly important in real-time IoT applications where delays can impact system performance.

Table 5.1 consolidates the above metrics, providing a clear comparative overview. The data affirm that while PUF authentication is more efficient in terms of resource utilization and speed, RC5 encryption provides the cryptographic strength necessary for data confidentiality. Their combination, although resource-intensive, delivers a balanced solution for secure and efficient IoT deployments and also summarizes the comparative performance of the two components in terms of latency, CPU usage, memory usage, and energy consumption.

Table 5.1: Detailed Efficiency Comparison of PUF Authentication, RC5 Encryption, and Combined Framework

| Metric | Authentication | RC5 Encryption | Combined |
|-------------------------|----------------|----------------|----------|
| Energy Consumption (mJ) | 2.5 | 4.8 | 7.3 |
| Memory Usage (KB) | 32 | 48 | 80 |
| CPU Usage (%) | 8 | 15 | 23 |
| Latency (ms) | 5 | 12 | 17 |

It can be observed from Table 5.1 that the PUF authentication takes much lesser energy and latency compared to the RC5 encryption. It is evident from the data that even though both the blocks are light-weight, the block of authentication is highly efficient with very low overhead. Even though RC5 encryption is very resource-consuming, it is yet appropriate for real-time data security in IoT applications. Figures 5.1 present the features of performance graphically, presenting a clearer difference between the authentication and encryption component. In summary, the experimental results validate that the new approach is extremely applicable to real-world IoT implementations. The combined use of PUFs for authentication and RC5 for encryption ensures robust security without introducing significant performance penalties. It addresses core challenges such as energy efficiency, computational simplicity, and resistance to physical and cyber threats.

Figure 5.2 presents a comparative analysis of two security methods — PUF Authentication and RC5 Encryption — across four key performance metrics: Energy consumption (in millijoules), Memory usage (in kilobytes), CPU Usage (in percentage), and Latency (in milliseconds). The x-axis categorizes these metrics, while the y-axis represents the measurement values, ranging from 0 to 50.

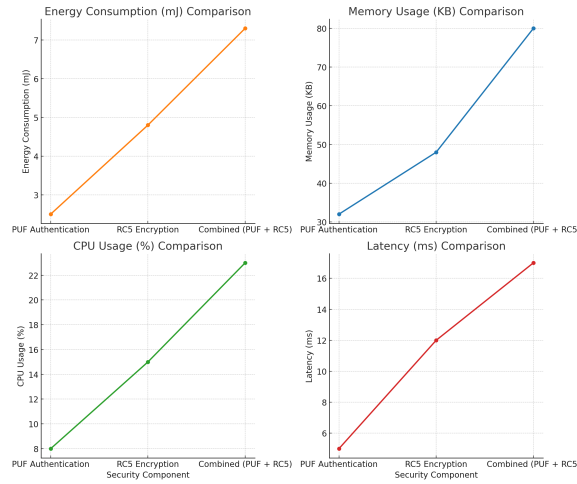


Figure 5.1: Comparison of Latency, Memory CPU and Energy

The PUF Authentication method, represented by a yellow-orange line with circular markers, shows a relatively low energy consumption of approximately 3 mJ, which is lower than RC5's energy usage of about 5 mJ. For memory utilization, PUF requires roughly 32 KB, significantly less than the 48 KB demanded by RC5 Encryption. CPU usage for PUF is around 8

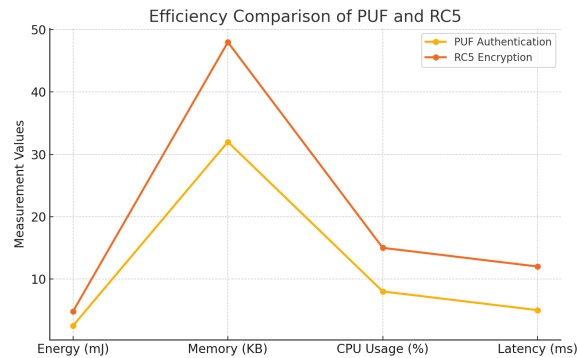


Figure 5.2: Efficiency Comparison

Across all four performance metrics — energy, memory, CPU usage, and latency. PUF Authentication demonstrates better efficiency with lower resource consumption and faster processing times. The RC5 Encryption method, indicated by a reddish-orange line with circular markers, consistently shows higher measurement values, suggesting it is more resource-intensive and slower in comparison. The clear visual distinction with color coding and markers, combined with the gridlines on the chart, allows for an intuitive comparison between these two security methods, highlighting the efficiency advantages of PUF Authentication over RC5 Encryption.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This dissertation suggests a secure architecture intended to safeguard strong authentication and data confidentiality for Internet of Things (IoT) networks. The main goal was to authenticate the legality of IoT sensor nodes and ensure the confidentiality and integrity of their shared data by adopting efficient cryptographic schemes. By integrating the light-weight RC5 encryption process with Physically Unclonable Functions (PUFs), the presented framework efficiently resolves major IoT security issues, including unauthorized access to devices and sniffing of data. PUFs provide in-security hardware-based authentication by creating device-specific, one-time fingerprints from the natural silicon manufacturing variation. With experimental results showing high reliability (greater than 95RC5, a simple and effective encryption algorithm, was chosen due to its aptitude to be implemented in resource-limited environments. It imposed less computational overhead, had lower memory demands, and was faster than available lightweight encryption algorithms. These features make RC5 an implementable solution for data confidentiality in low-power IoT networks. Experimental results certified the effectiveness of the framework in device authentication and data protection without utilization of high resources consumption. The proposed solution proved to be scalable and adaptable for diverse IoT deployment scenarios, ensuring secure communication without compromising performance [28]. Overall, the integration of PUFs and RC5 encryption presents a secure, efficient, and flexible approach for addressing modern IoT security requirements.

6.2 Future Work

Although the proposed framework demonstrates substantial advantages in terms of performance, scalability, and security, several promising avenues remain open for future research to further refine and extend its capabilities. One potential direction involves the integration of advanced error correction codes (ECC) to enhance the reliability of PUF responses under varying environmental conditions. Sophisticated ECC techniques could ensure the consistency and robustness of cryptographic key generation, particularly in harsh operational settings where traditional methods may falter. The research on hybrid encryption protocols is another rich extension. Merging symmetric encryption schemes like RC5 with asymmetric cryptography techniques could possibly yield more secure means of key exchange, particularly for large-scale and distributed IoT networks

where heterogeneous devices must securely communicate among themselves.

Furthermore, the use of machine learning methods offers a robust potential to enhance security monitoring and vulnerability scanning. Employing data analysis in real-time, machine learning algorithms would be capable of continuously evaluating threats and modify security settings accordingly, thus further enhancing the resilience and intelligence of the overall IoT security system. These recommendations overall provide an outline for transforming the existing solution into a more holistic and wise architecture in a position to fulfill the sophisticated needs of next-generation IoT systems.

Bibliography

- [1] D. Pishva, “Internet of things: Security and privacy issues and possible solution,” in *International Conference on Advanced Communication Technology (ICACT)*, Bongpyeong, South Korea, 2017.
- [2] D. Mukhopadhyay, “Pufs as promising tools for security in internet of things,” *IEEE Design Test*, 2016.
- [3] B. Halak, M. Zwolinski, and M. S. Mispan, “Overview of puf-based hardware security solutions for the internet of things,” in *IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS)*, Abu Dhabi, United Arab Emirates, 2017.
- [4] S. Horrow and A. Sardana, “Identity management framework for cloud based internet of things,” in *Proceedings of the First International Conference on Security of Internet of Things*, Kollam, India, 2012.
- [5] R. Vilalta, R. Ciungu, A. Mayoral, R. Casellas, R. Martinez, D. Pubill, J. Serra, R. Munoz, and C. Verikoukis, “Improving security in internet of things with software defined networking,” in *IEEE Global Communications Conference (GLOBECOM)*, Washington, DC, USA, 2016.
- [6] J. Liu, Y. Xiao, and C. L. P. Chen, “Authentication and access control in the internet of things,” in *32nd International Conference on Distributed Computing Systems Workshops*, Macau, China, 2012.
- [7] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, “A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications,” *IEEE Internet of Things Journal*, vol. 4, no. 5, 2017.
- [8] N. M. Kumar and P. K. Mallick, “The internet of things: Insights into the building blocks, component interactions, and architecture layers,” *Procedia Computer Science*, vol. 132, pp. 109–117, 2018.
- [9] N. M. Lobanchykova, I. A. Pilkevych, and O. Korchenko, “Analysis of attacks on components of iot systems and cybersecurity technologies,” in *CEUR Workshop Proceedings*, 2021, pp. 83–96, in press.
- [10] Z. Liao, S. Nazir, H. U. Khan, and M. Shafiq, “Assessing security of software components for internet of things: a systematic review and future directions,” *Security and Communication Networks*, vol. 2021, pp. 1–22, 2021.
- [11] C. M. M. Otalvaro, J. C. B. Andrade, C. M. Z. Jaramillo, and J. I. RiosPatiño, “Iot best practices and their components: A systematic literature review,” *IEEE Latin America Transactions*, vol. 20, no. 10, pp. 2217–2228, 2022.

- [12] F. Molaei, E. Rahimi, H. Siavoshi, S. G. Afrouz, and V. Tenorio, "A comprehensive review on internet of things (iot) and its implications in the mining industry," *American Journal of Engineering and Applied Sciences*, vol. 13, no. 3, pp. 499–515, 2020.
- [13] S. Bansal and D. Kumar, "Iot ecosystem: A survey on devices, gateways, operating systems, middleware and communication," *International Journal of Wireless Information Networks*, vol. 27, pp. 340–364, 2020.
- [14] C. Patel and N. Doshi, "Security challenges in iot cyber world," in *Security in Smart Cities: Models, Applications, and Challenges*. Springer, 2019, pp. 171–191.
- [15] M. Litoussi, N. Kannouf, K. E. Makkaoui, A. Ezzati, and M. Fartitchou, "Iot security: challenges and countermeasures," *Procedia Computer Science*, vol. 177, pp. 503–508, 2020.
- [16] J. Mohanty, S. Mishra, S. Patra, B. Pati, and C. R. Panigrahi, "Iot security, challenges, and solutions: a review," in *Progress in Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2019, Volume 2*. Springer, 2021, pp. 493–504.
- [17] M. Tahir, M. Sardaraz, S. Muhammad, and M. Saud Khan, "A lightweight authentication and authorization framework for blockchain-enabled iot network in health-informatics," *Sustainability*, vol. 12, no. 17, p. 6960, 2020.
- [18] R. Krishnamurthi, A. Kumar, D. Gopinathan, A. Nayyar, and B. Qureshi, "An overview of iot sensor data processing, fusion, and analysis techniques," *Sensors*, vol. 20, no. 21, p. 6076, 2020.
- [19] C. Wheelus and X. Zhu, "Iot network security: Threats, risks, and a data-driven defense framework," *IoT*, vol. 1, no. 2, pp. 259–285, 2020.
- [20] Z. A. Baig, S. Sanguanpong, S. N. Firdous, T. G. Nguyen, and C. So-In, "Averaged dependence estimators for dos attack detection in iot networks," *Future Generation Computer Systems*, vol. 102, pp. 198–209, 2020.
- [21] Y. Sung, S. Lee, and M. Lee, "A multi-hop clustering mechanism for scalable iot networks," *Sensors*, vol. 18, no. 4, p. 961, 2018.
- [22] K. Echenim, L. Elluri, and K. Joshi, "Ensuring privacy policy compliance of wearables with iot regulations," UMBC Center for Accelerated Real Time Analysis, Tech. Rep., 2023.
- [23] A. J. Perez, S. Zeadally, and J. Cochran, "A review and an empirical analysis of privacy policy and notices for consumer internet of things," *Security and Privacy*, vol. 1, no. 3, p. e15, 2018.
- [24] P. Morgner and Z. Benenson, "Exploring security economics in iot standardization efforts," *arXiv preprint arXiv:1810.12035*, 2018.
- [25] R. Mishra and R. Yadav, "Access control in iot networks: analysis and open challenges," in *Proceedings of the International Conference on Innovative Computing & Communications (ICICC)*, March 2020.

- [26] Y. Jiang, C. Wang, Y. Wang, and L. Gao, “A cross-chain solution to integrating multiple blockchains for iot data management,” *Sensors*, vol. 19, no. 9, p. 2042, 2019.
- [27] S. Ravidas, A. Lekidis, F. Paci, and N. Zannone, “Access control in internet-of-things: A survey,” *Journal of Network and Computer Applications*, vol. 144, pp. 79–101, 2019.
- [28] K. Riad and J. Cheng, “Adaptive xacml access policies for heterogeneous distributed iot environments,” *Information Sciences*, vol. 548, pp. 135–152, 2021.
- [29] S. R. Oh, Y. G. Kim, and S. Cho, “An interoperable access control framework for diverse iot platforms based on oauth and role,” *Sensors*, vol. 19, no. 8, p. 1884, 2019.



DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

PLAGIARISM VERIFICATION

Title of the Thesis Authentication of IOT devices using PUF and Encrytion Technique

Total Pages 40 Name of the Scholar Mohit Sharma

Supervisor (s)

(1) Dr. Sanjay Patidar

(2) _____

(3) _____

Department Department of Software Engineering

This is to report that the above thesis was scanned for similarity detection. Process and outcome is given below:

Software used: Turnitin Similarity Index: 9% Total Word Count: 10,230

Date: 30/5/25

M/sharma
Candidate's Signature

[Signature]
Signature of Supervisor(s)

Authentication_of_IOT_Devices_using_PUF_&_Encryption.pdf



Delhi Technological University

Document Details

Submission ID

tmsid::27535:97177898

Submission Date

May 22, 2025, 2:40 PM GMT+5:30

Download Date

May 22, 2025, 2:57 PM GMT+5:30

File Name

M_Tech_Thesis_final_final_final9768987987.pdf

File Size

5.2 MB

40 Pages

18,230 Words

64,952 Characters

9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 8 words)

Match Groups

- 66 Not Cited or Quoted 9%
Matches with neither in-text citation nor quotation marks
- 0 Missing Quotations 0%
Matches that are still very similar to source material
- 0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

- 6% Internet sources
- 5% Publications
- 6% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Authentication_of_IOT_Devices_using_PUF_&_Encryption.pdf

 Delhi Technological University

Document Details

Submission ID

trnoid::27535:97177898

Submission Date

May 22, 2025, 2:40 PM GMT+5:30

Download Date

May 22, 2025, 2:57 PM GMT+5:30

File Name

M_Tech_Thesis_final_final_final9768987987.pdf

File Size

5.2 MB

40 Pages

10,230 Words

64,952 Characters



0% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

-  **1 AI-generated only 0%**
Likely AI-generated text from a large-language model.
-  **0 AI-generated text that was AI-paraphrased 0%**
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate. It is not a substitute for writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated. It should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

