DESIGNING A HYBRID LIGHTWEIGHT LOGICAL SECURITY FRAMEWORK FOR INTERNET OF THINGS

A

Thesis

Submitted to

For the award of

DOCTOR OF PHILOSOPHY (Ph.D)

in

COMPUTER SCIENCE AND ENGINEERING

Submitted By

Isha Batra

11412467

Supervised By

Dr. Sahil Verma

LOVELY FACULTY OF TECHNOLOGY AND SCIENCES
LOVELY PROFESSIONAL UNIVERSITY
PUNJAB
2019

CANDIDATE'S DECLARATION

I hereby declare that the thesis entitled, **DESIGNING A HYBRID LIGHTWEIGHT LOGICAL SECURITY FRAMEWORK FOR INTERNET OF THINGS** submitted for the Degree of Doctor of Philosophy in Computer Science and Engineering is the result of my original and independent research work carried out under the guidance of Supervisor Dr. Sahil Verma, Assistant Professor, School of Computer Science and Engineering, Lovely Professional University, Punjab. This work has not been submitted for the award of any degree, diploma, associateship, fellowship of any University or Institution.

Date: **Investigator:** Isha Batra

CERTIFICATE

This is to certify that the thesis entitled *“Designing a Hybrid Lightweight Logical Security Framework for Internet of Things”* submitted by Isha Batra for the award of degree of Doctor of Philosophy in Computer Science and Engineering, Lovely Professional University, is entirely based on the work carried out by her under my supervision and guidance. The work reported, embodies the original work of the candidate and has not been submitted to any other university or institution for the award of any degree or diploma, according to the best of my knowledge.

Signature of Advisor

Name: Dr. Sahil Verma

Date:

ABSTRACT

Internet of Things (IoT) is defined as a network of uniquely identifiable, accessible, and manageable smart things that are capable of communication, computation and ultimate decision making. Smart things can be any physical objects like phone, laptop, refrigerator, AC, charger, and many more. IoT is an emerging technology in this expanding era of smart things. Things in IoT can be connected using wireless connections like RFID, Bluetooth, ZigBee, WSN, WLAN, WMAN, or Wi-Fi. The number of things or users connected to IoT is growing exponentially and around 2020 the number of connections may reach 50 billion. Hence, the bandwidth requirement of IoT will also increase exponentially. Licensed and unlicensed bands are available for communication. Licensed bands are paid and are used in applications like 2G, 3G, and many more. Unlicensed frequency bands are reserved for industrial, scientific and medical applications also known as ISM bands. The existing ISM bands are 433 MHz, 915 MHz and 2.4 GHz. The ISM band used in IoT is 2.4 GHz for Wi-Fi enabled communication.

IoT helps in creating connections between dissimilar things present in heterogeneous environment. This kind of openness and very less human intervention can make IoT exposed to number of attacks like man in middle attack, Denial of service (DoS) attack. Moreover, any device can access the network that leads to unauthorized access. These attacks can damage device physically and network connections too. This will ultimately compromise the security and privacy of IoT. As, IoT is resource constrained with less power, bandwidth, less storage, so an efficient security solution is required that will not chomp through the resources of IoT.

Therefore, the primary groundwork of this research work is to maintain the trust of users in IoT by securing data from unauthorized reading and modification, and preserving privacy of the users. To achieve this, the entire research works is carried to (1) propose a new hybrid lightweight logical security framework to assure confidentiality and integrity of information with optimal use of resources, (2) compare proposed hybrid lightweight logical security framework with the existing frameworks in terms of their key size, key distribution mechanism, execution time, energy

efficiency, memory requirements, and (3) validate the proposed framework towards various attacks and evaluate performance in terms of throughput, latency, and packet delivery ratio.

In the way of achieving the first objective, that is to propose a new hybrid lightweight logical security framework to assure confidentiality and integrity of information with optimal use of resources. First the existing security algorithms are discussed in IoT. Conventional cryptographic algorithms used in literature can be symmetric or asymmetric. Examples of symmetric algorithms are AES, RC5, HIGHT, TEA, PRESENT, and many more Examples of asymmetric are RSA, ECC. The conventional cryptographic algorithms are not apt for the IoT scenarios as IoT is resource constrained. Later, researchers proposed another category of lightweight cryptographic algorithms. Lightweight refers to algorithms working on light requirements for execution like memory, time. Lightweight algorithms are well suited for constrained environments as they performs with same level of security as of conventional algorithms but requires less execution time, and have less processing requirements. A tradeoff is always maintained between security and performance in case of lightweight algorithms.

Literature study claims that the existing lightweight solutions for IoT are vulnerable to attacks like Denial of Service, Differential attack, related key attack. To further get an optimized security solution for IoT, a Hybrid Lightweight Security Framework ((HLSF) is proposed in this research work that offers authentication, confidentiality, and integrity. The framework is divided into three phases. First phase is registration where every device that connects to the network gets the credentials from the server. Second phase is authentication that ensures validity of devices connecting to IoT. Once the devices are authenticated algorithm proceeds to the last phase. Third phase works on securing the data in transit when data is transmitted from one end to other. A new Lightweight Data Security (LDS) algorithm is proposed for securing data that offers confidentiality and integrity of data.

Contiki operating system comes with a sophisticated IoT Simulator, COOJA, which is based on real hardware emulation and time-accurate IoT simulation. Existing lightweight solutions like SPECK, SIMON, FANTOMAS, TWINE, and proposed LDS are compared using COOJA as the simulator. The comparison is made on the basis of memory requirement, throughput, energy efficiency, avalanche effect and execution time. The results of comparison show that out of existing lightweight solutions for IoT, LDS performs relative well with minimum execution time and memory requirements and maximum throughput. Later, comparison is made between two variants of SPECK and the LDS security algorithm of the proposed framework considering varying key that is 96 bits and 128 bits and block size of 64 bits. The evaluation results show that the proposed security algorithm is performing well in terms of execution time, memory requirements and throughput.

The security solution for IoT should be an optimum one in terms of above stated performance parameters in trade off with the level of security. The proposed security algorithm is performing quite well than SPECK but a security check need to be done on both the algorithms. Security check refers to do cryptanalysis on proposed algorithm and SPECK. Literature shows that SPECK is vulnerable to number of attacks like differential attack, 2-Round attack, Rectangle Attack. This research work performs cryptanalysis of proposed algorithm on differential attack. The results shows that SPECK is vulnerable to differential attack at 19 out of 27 rounds. The Proposed algorithm is vulnerable to differential attack at 10 out of 20 rounds. The attack complexity of SPECK is 2^{125} and that of LDS is evaluated as 2^{127}, leading more time to break LDS. Finally, security level of SPECK and LDS is calculated. Considering all performance parameters and security level, the proposed algorithm LDS outperforms than SPECK.

For evaluating the proposed HLSF as whole, it is compared with existing security frameworks CoAP and OSCAR. Simulation scenario having 30-100 motes is created using COOJA. The energy overhead, computation overhead, memory requirements, throughput, packet delivery ratio, and latency is evaluated for all CoAP, OSCAR, and HLSF including all the phases of authentication, and data security using the same underlying technology. Evaluation results shows that HLSF is more energy efficient that is having less computational overhead, less energy overhead, requires less memory requirements, posses high throughput and packet delivery ratio with minimum latency out of all.

IoT application starts with data collection, data security and then data mining that will help in taking final decision. For data collection LEACH protocol is used, for

data security HLSF is proposed and Frequent Pattern (FP) association based mining is used as the data mining approach in this research work. Considering the application scenario of inventory management system in IoT, the overall performance of HLSF and CoAP is compared assuming same data collection method and data mining approach. The performance is compared on the basis of precision, recall, and accuracy. Result shows that HLSF when used in IoT scenario gives relatively high precision, recall and accuracy in decision making as compared to existing frameworks.

ACKNOWLEDGEMENT

I would like to present my deepest gratitude to **Dr. Sahil Verma** for his guidance, advice, understanding and supervision throughout the development of this thesis and study. Despite his busy schedule he has been available at every step, devoting time and energy and the much needed counsel and advice. This enabled me to sail through the tough times and complete this enormous task.

I would like to thank to the **research project committee members** for their valuable comments and discussions. A special thanks to the management of **Lovely Professional University** for their support in academic concerns and letting me involve in research study. The doctoral programme of LPU has made it possible for me to pursue my dream of research and upgrade my knowledge.

My sincere feeling of gratefulness also goes to my parents and family members who always motivated me in all the endeavours of my life including this research work in LPU. I am also thankful to my husband Dr. Arun Malik for offering full support to me during the entire period of my research work. My special thanks to mu daughter Prisha for giving me joyful and happy moments during the entire journey of my research work. Finally, I would like to thank each and every person who has directly and indirectly helped and motivated me in this journey.

ISHA BATRA

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

Abbreviation	**Description**
6LBR	6LoWPAN Border Router
6LoWPAN	IPv6 Low Power Wireless Personal Area Network
AES	Advanced Encryption Standard
ARX	Addition, Rotation and XOR
BS	Base Station
CA	Certificate Authority
CBC	Cipher Block Chain Mode
CCM	Cipher Block Chain Mode
CoAP	Constrained Application Protocol
DES	Data Encryption Standard
DH	Diffie-Helmen
DoS	Denial of Service
DTLS	Datagram Transport Layer Protocol
ECC	Elliptic curve cryptography
ECDHE	Elliptic Curve with Diffie-Helmen and ephemeral key
EPoSS	European Technology Platform on Smart System integration
ER	Effective Rate
FHSS	Frequency Hopping Spread Spectrum
FP	Frequent Pattern
GPS	Global Positing System
GUI	Graphical User Interface
HIGHT	High Security and Lightweight
HLSF	Hybrid Lightweight Security Framework
HTTP	Hyper Text Transfer Protocol
IANA	Internet Assigned Numbers Authority
ICT	Information and Communication Technology
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IMEI	International Mobile Equipment Identity
IoT	Internet of Things
ISO	International Standard Organization
LDS	Lightweight Data Security
LEACH	Low Energy Adaptive Clustering Hierarchy
LPS	Local Position System
ML	Machine Learning
NIST	National Institute of Standards and Technology
OSCAR	Object Security Framework
Pr	Precision
QoS	Quality of Service
RAM	Random Access Memory
Re	Recall
RFID	Radio frequency Identification
ROM	Read Only Memory
RPL	Routing Protocol for Low Power and Lossy networks

SIS	Smart Inventory System
TC	Time Complexity
TCP	Transmission Control Protocol
TEA	Tiny Encryption Algorithm
TLS	Transport Layer Security
UDGM	Unit Disk Graph Medium
UDP	User Datagram Protocol
URL	Uniform resource locater
URN	Uniform Resource Name
UWB	Ultra Wide Band
VoIP	Voice Over IP
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network

CHAPTER 1

INTRODUCTION

This section builds up the knowledge required to understand the problem statement of the research work by understanding the basic working of Internet of Things (IoT), components required for shaping IoT, its applications, challenges in operation of IoT, IoT security measures through layered architecture, and the possible attacks on each layer and their available solutions. Based on the existing set of problems, multiple research objectives have been framed to carry out this research work.

1.1 Introduction

With the expansion in number of items or gadgets that are associated with web, utilizing them in genuine globe to discuss between themselves and make ponders, Kevin Ashton, individual from Radio Frequency Identification (RFID) cited the term Internet of things (IoT) in Forbes magazine [1]. IoT is a system framed from physical things by and large alluded as items that speak with one another in standardized way. Objects can be anything that we come across every day and are connected through internet to user application as shown in Figure 1.1. Objects are necessarily not electronic gadgets; but can be day by day used gadgets like paper, pen, key, garments, etc. In [2] writer referenced that the articles can be discernible, effectively available, locatable, and tended to by utilizing any methods for correspondence like web, LAN, WAN, or RFIDs, or sensors.

Using the features of wireless networks, the main intent of IoT is on gathering the information from users who are authorized in the system. According to a survey, it has been estimated around 50 billion smart things are going to part of IoT community by 2020 [3].Various organizations and working groups are focused on IoT today like Samsung, apple, Thread group, etc. According to press release by Dr. Hong, Samsung have already launched the smart home kits which work on different modules using same application [4]. For example, it will monitor the lights, or AC, or refrigerator, if no one in the room it will automatically switch off the lights and AC. If someone breaks the door or glass of home it raises the alarm to the application. To achieve this motion sensors are used to check the movement of user.

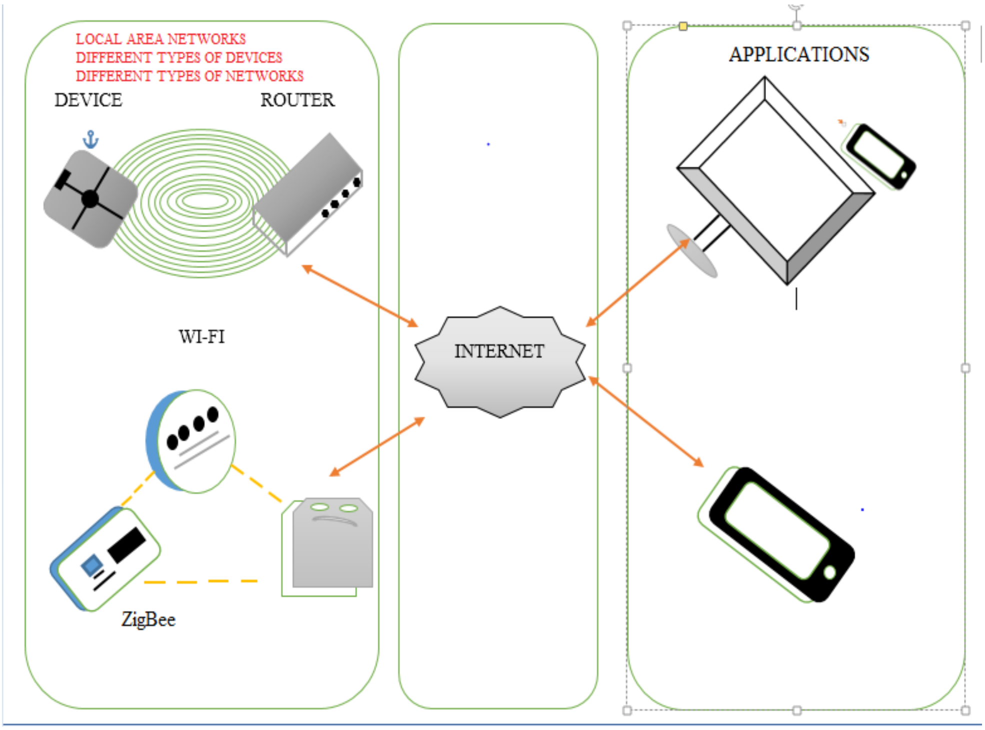


Figure 1.1: IoT Scenario

In [5] the most conspicuous working application zones of IoT can be distinguished as observing condition, medicinal services the board, mechanized savvy homes, industry, retail, inventory network the board, coordination, horticulture, and some more. IoT scenario is well suited for environments where real time communication is maintained among the objects or things on diverse platforms.

With innovation in IoT, quantities of things associated with it are extending, in this way the bandwidth requirements increments. Frequency band can be authorized or unlicensed. Application such as 2G or 3G, make use of authorized bands. For other specific applications like scientific, industrial or medical, frequency bands that are unlicensed referred as ISM bands are used. These bands available are 2.4 GHZ, 915 MHz, and 433 MHz. For communication that is Wi-Fi enabled in IoT, ISM band of 2.4 GHz is used [6]. As indicated by a report by European commission [7], the eventual fate of IoT is determined in distinct phases by European Technology Platform on Smart System incorporation (EPoSS). For 2010, retail and coordination, for 2010-2015, associating objects, for 2015-2020, fractional keen articles, after 2020, intelligent working items.

1.2 Components Required for Shaping IoT

IoT is not a single module, but it consists of multiple other components. This section presents the different components that form the whole IoT network scenario.

1.2.1 Radio Frequency identification (RFID)

RFID comprise of electronic standardized identifications that is utilized to remarkably recognize anything to which it is connected. It works on wireless communication technology through the use micro chips. RFID is a better solution over existing 1D and 2D barcodes technologies as it allows the updating of data in the chip. RFID operates from very low frequency band of around 125 KHz to very high frequency band of 5.8 GHZ [8]. It do not require any intervention of human and can be used to recognize the fast moving objects and in difficult environments also.

RFID essentially comprises of two principle parts-a tag and a reader as appeared in Figure 1.2. Reader is utilized to get to the prepared data in tag and impart it to the application framework

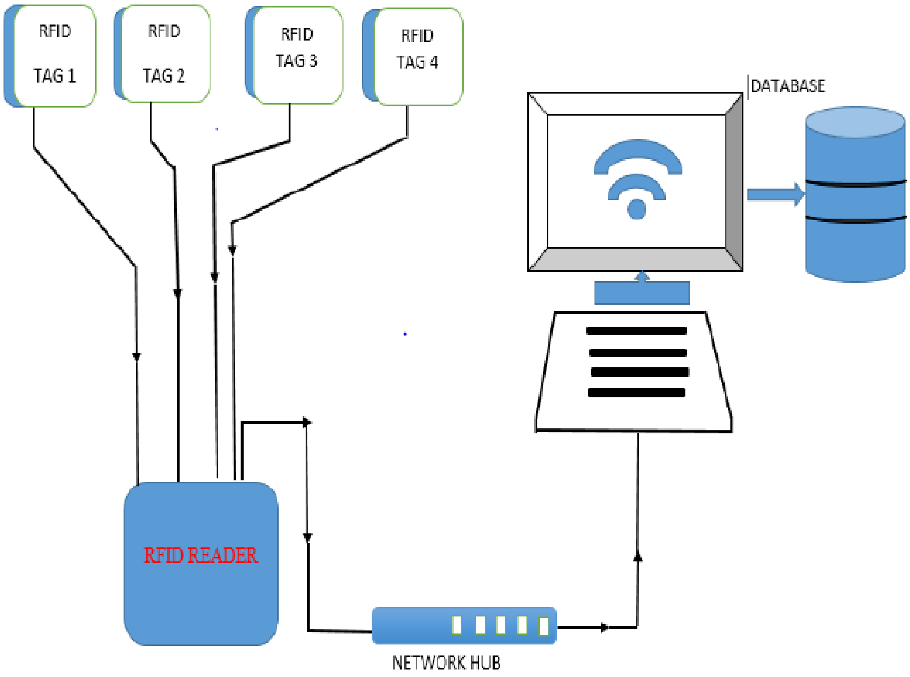


Figure 1.2: Working Principle of RFID

a) RFID Tag

Tag comprises of a microchip that is used to store the object's unique identity to which it is connected and other data related to the object. An antenna is also attached that is meant for two way communication among the tag and the reader. RFID labels might be active, passive or partially passive. A tag that sends the data occasionally to the reader and is battery worked is Active tag. If a tag only sends the information when required by the receiver and is working on small battery, it is partially passive. On the other hand, a tag having no battery and works on energy gained from receiver, is passive. The difference among different types of tags is discussed in Table 1.1 below.

Table 1.1: Comparison of RFID Tags

Property	**Active Tag**	**Passive Tag**	**Partial Passive Tag**
Power Source	Internal	From signal received	Internal
Life time	Less	More	More
Memory	Large	Small	Large
Sensor	Yes	No	Yes
Range	Above 100 meters	Few meters	Above 100 meters
Example	Military cargo	ATM card/Toll road	Inventory

b) RFID Reader

It is used to inquire about any information about an object. In [10] it is described that reader transfers a message to RFID tag to access stored information. The accessed information is then communicated to the outer application system for decision making. RFID reader can also be an active reader or passive reader. An active reader can communicate to/from with the active tag as well the passive tags for gathering the required information. A passive reader can only receive the information from the active tags. RFID specification can be either from EPC or International Standard Organization (ISO)

Some of the inventory stores are using barcodes or RFIDs for billing system or for security purpose. With the existing RFIDs tag enabled items in inventory, this era of smart phones can be better utilized to make the inventory system smarter. Technologies required for the execution of Smart Inventory System (SIS) are RFID tag, RFID readers, interconnected IoT system using Wi-Fi, and based on Wi-Fi a Local Position System (LPS). The SIS will improve the traditional inventory system and will tend to provide the following facilities also-

- To manage the inventory system by maintaining current stock information on server for each item, to avoid overstock and outages.
- To authenticate a customer through his smart phone.
- To find the current location of the item and direct the customer to that particular location using LPS.
- To update the server if any item is added or removed from the shelf or even if the whole new shelf is added, moved or removed from the store.

RFID is used for short range communication and is apt for the inventory management system consists of tags and readers. RFID maintains the information regarding an item in tag in form of chip attached to the item. This information is passed to the reader on receiving a RF signal [11].

Once the information is collected by the shelf monitor, it is updated in the server. User queries the server for a particular item. If item is present, server responds to the customer with the shelf number. Another technology used by the algorithm is LPS. LPS is used to find the location inside the buildings or areas with no clear line of sight. LPS makes use of networks that are wireless or Wi-Fi to send beacons to track the position. LPS will help the customer to locate the item based on the shelf number received from server.

1.2.1.1 Smart Inventory System

In Inventory management system each item has a RFID tag incorporated in item. Tag maintains a unique code for each item and with the help of RFID reader this information is read. The main purpose of RFID tag and reader is in billing or for security check so that an item cannot be taken outside the store and security is required for e-commerce applications as well based on SOA [12].

The existing inventory system is extended to make it smarter, by using other technologies like Wi-Fi, and a LPS [13]. The architecture of SIS as shown in Figure 1.3 comprises of server, shelf monitor for each shelf holding items, Wi-Fi enabled LPS, and smart phone of user.

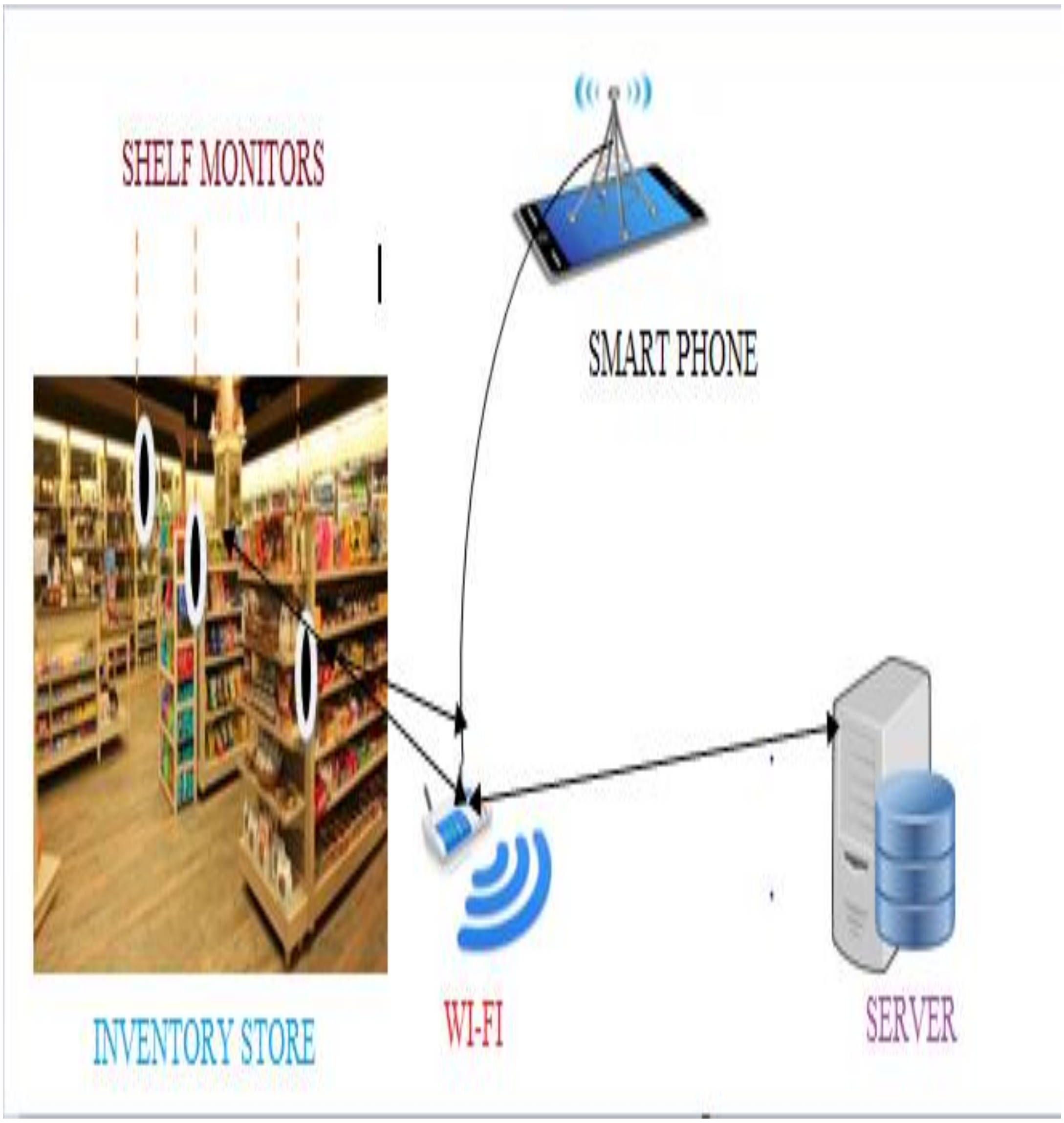


Figure 1.3: Architecture of SIS

The following user scenario describes the use of IoT in application of SIS and the flow of data is represented in Figure 1.4.

- Every new customer have to register with the inventory system for the first time with his smart phone with unique International Mobile Equipment Identity (IMEI). On registration, customer will be provided a username and a password.
- Existing customer will enter his username and password for authentication. Device authentication is achieved by matching the IMEI number stored in database.
- Customer enquires the server about a particular item in the inventory store. For this user enters the item name and corresponding brand name. For example-juice will be entered as brand name and Tropicana will entered in brand name.
- If item is available, server will respond to the smart phone of the customer regarding the shelf number of the item. On receiving the shelf number, smart phone will contact the LPS using Wi-Fi to get the exact position where the item is placed.
- Customer will follow the instructions by tracking the location provided by the application.
- If item is not available in the inventory, the server will respond with a message that no such item found. Customer may then enter name of any other he wants to search for.
- Shelf monitor sends continuous updates the server with the current status of items present in a particular shelf. Item Updating can be due to if any item is deleted, added or its quantity get changed.

1.2.1.2 Components of SIS

As shown in Figure 1.3 the SIS, comprises of smart phone, shelf monitors, server, and Wi-Fi based LPS. The main motive of providing this smart system is to help the customers to get the exact status and location of a particular item in the inventory store. In this searching process the various component modules that are required for customers are as follows:

- **Authentication:** Authentication is a twofold process. One is the customer authentication and other is the smart phone authentication. When customers enter the inventory store to locate the position of a item, he has to first use the application and enter his username and password. The login request goes to server. Server validated the customer if he has entered the same username and password as maintained in the database. Server will also guarantee the validity of the device by matching the IMEI number with the one that is stored IEMI for that particular customer.

- **Query to Server:** Each item in the inventory store is equipped with an RFID tag. The details of items in one shelf are maintained by a shelf monitor having a RFID reader. Whenever a new item is added on deleted from shelf, it is notified to shelf monitor and by the use of Wi-Fi, shelf monitor communicates this updated information to the server. When customer wants to search a particular item he has to enter his name and the brand of that item he is searching. The request is sent to the server if item is present in the store its shelf number and its quantity are returned by the server.

- **Searching an Item:** Using the LPS of smart phone, customer can track the exact location of the item. LPS uses the customer coordinates and the shelf coordinates to find the location of the item in the inventory store.

- **Updating the Server:** Server updating is required in three ways. Firstly, when a item is added or deleted in a particular shelf its status is updated. Secondly, when a shelf itself it shifted, added or deleted from the store its information is updated by the LPS in the server. Lastly, when customer purchases a particular item its status is updated in the database.

Figure 1.4 shows the full user scenario offered for SIS discussing how the status of items in the inventory can be automatically received. Based on the outcome of SIS, inventory stores can be maintained in much easy manner.

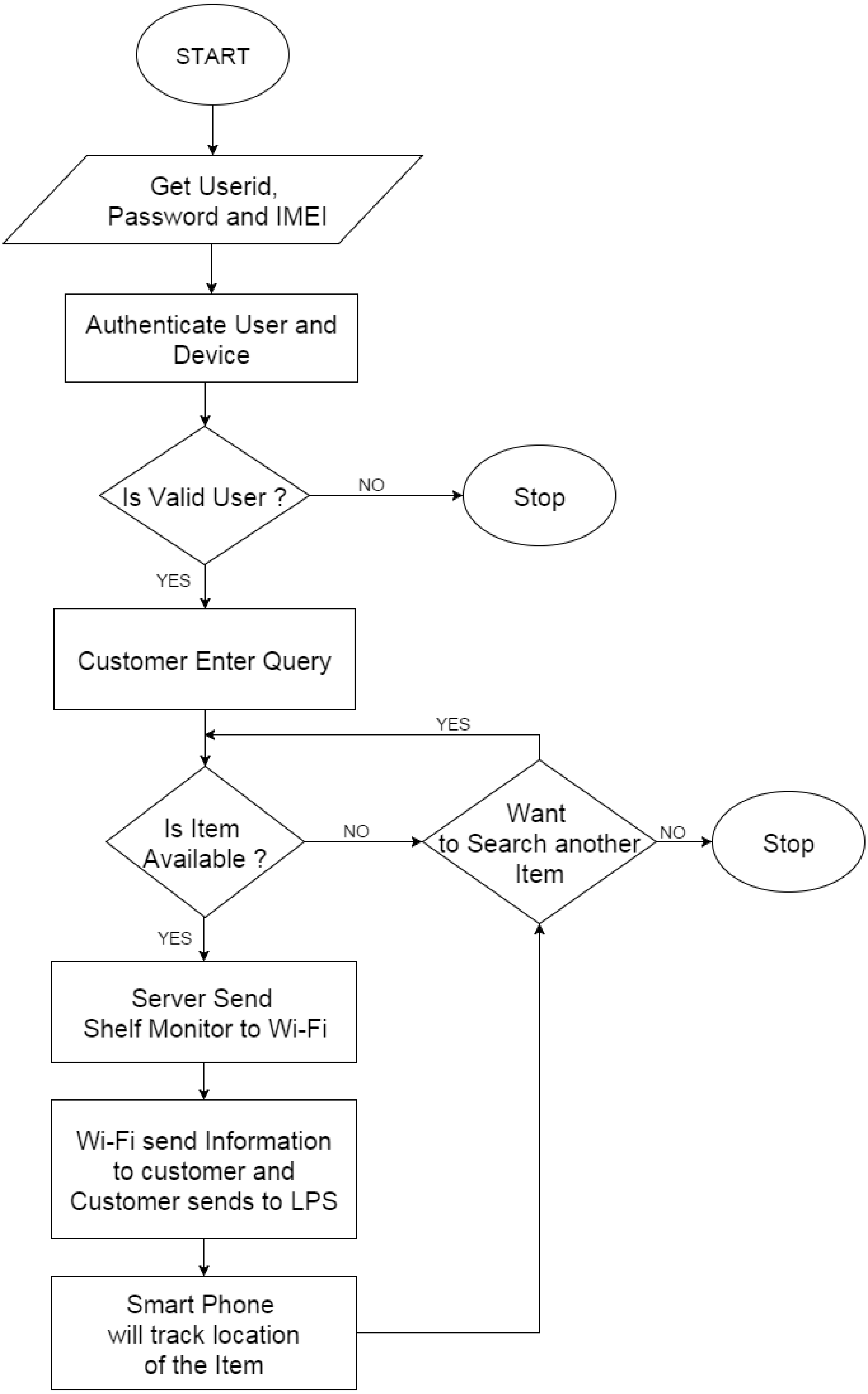

START
Get Userid,
Password and IMEI
Authenticate User and Device
Is Valid User ?
NO
Stop
YES
Customer Enter Query
YES
Is Item Available ?
NO
Want to Search another Item
NO
Stop
YES
Server Send Shelf Monitor to Wi-Fi
Wi-Fi send Information to customer and Customer sends to LPS
Smart Phone will track location of the Item

Figure 1.4: Flowchart Representing User Scenario for SIS

1.2.2 Wireless Sensor Network (WSN) technology

In [14] it is depicted that a WSN comprises of different low power battery worked sensor hubs that are dispersed in arbitrary style. A sensor node comprise of sensing units, transceiver, processing and storing unit and battery. After collecting information from the environment, sensor nodes pass this information to some special nodes called sink nodes. Sink nodes further pass this collected information to the internet. Sensor networks face different challenges like scalability (change of number of nodes), energy efficiency (as nodes are battery operated), reliability (nodes can die after some time). So the research work in sensor networks is concentrated on MAC layer, routing, security in data aggregation.

1.2.3NetworkAddressing and mapping schemes

Due to growth in the number of things operating in IoT, the need of identifying them uniquely on the network is also increasing [15]. Internet Assigned Numbers Authority (IANA) had already allocated most of the IPv4 addresses and the allocated pool is almost exhausted. Now to facilitate communication, IPv6 is going to replace IPv4 having 3.4×10^{38} unique addresses. So, IPv6 is used in IoT to uniquely identify each thing related to the network. But for the mapping of external gateway and sensor network, Uniform resource name (URN) is used for addressing of sensor nodes. Each sensor node is provided a unique URN address through which the gateway can communicate with it. Moreover, a Uniform resource locater (URL) is also given to each sensor node for accessibility.

- **Network:** Wireless personal area network (WPAN) is used to establish communication among the things. It is a short range communication having low cost, less power, and low bit rate. Due to heterogeneity of devices a common communication platform is established among the things. The different mechanisms followed for medium access and physical layer are

- **IEEE 802.15.4:** Considered as a standard specification given by IEEE 802.15 task force group 4. Devadiga in [16] described that the physical and medium access control layer protocols of IoT are specified by it. Other standards like zigbee can be used which takes the IEEE 802.15 standard as the base and in addition specifies the upper layer protocols also. IEEE 802.15.4 standard work on three information rates of 250, 20, and 40 kbps that take a shot at the recurrence of 2.4 GHz, 868 MHz, and 915 MHz separately.

- **Ultra Wide Band (UWB):** UWB is used for communication of short transmission range but of high power rate. For example it can be used in transmission of audio, video, or in applications like tracking of exact location of a thing using sensors. UWB uses an impulse Radio technique to send the signals of very less duration to avoid larger duty cycles. So it overcomes the problems of interference, collision or multi fading that can be caused due to other methods like zigbee. UWB works on3.1 GHz up to 10.6 GHz range of frequencies.

- **Bluetooth:** A Bluetooth is a cable replacement technology as wireless LAN, but it is better than WLAN in terms of power and cost. The range of communication of Bluetooth is within 10 meters of range. Bluetooth works on ISM 2.402 GHZ on set of 79 channels where each channel occupies 1MHZ. it uses frequency hopping spread spectrum (FHSS) on 2.404 GHZ + k MHZ, where k value lies from 0 to 79.

  With hopping rate of 1600 hops/ second, Bluetooth creates an ad hoc network consisting of devices also called as piconet. A piconet is formed by connecting up to 7 slaves with 1 master. Master finds pattern of hopping and the slaves then synchronize accordingly. A unique hopping pattern is followed by each piconet. The benefit of Bluetooth is its less power consumption and minimum cost but due to its short distance range it cannot be used in long distance communications.

1.2.4 Intelligent Systems

All the above components discussed are used for data collection by themselves or through communication. But a system is required for processing the data collected and to apply that processed data for decision making in real world applications. As reported in by Atzori et al. in [17] the examples of such systems can be data mining, machine learning, intelligent control technology and human machine interaction system. All these systems can intelligently analyze the final decisions on the data collected.

1.3 Architecture of IoT

The connected devices in IoT are heterogeneous in nature so a standard architecture is required that will allow the intact communication among the devices of different types [18]. So the architecture of IoT can be viewed in 4 layers

- **Perception Layer/Physical Layer**: Acting as base layer in IoT it is used for actual information gathering by locating the things or objects. The devices used at perception layer are RFID readers/writers, sensors, GPS, RFID sensors. These devices are capable of collecting the information. The function of perception layer is data collection and to act as an informative layer. IEEE 802.15.4 takes a shot on MAC and physical layers for low speed, less fueled, shoddy systems.

- **Network Layer**: The layer above perception layer is the network layer. Principle capacity of network layer is transmitting the information gathered from perception layer. IPv6 addresses are used at network layer to identify the things uniquely. Network layer have features of information collection centre, management center, and transmission center.

- **Transport Layer**: This layer is utilized to make a start to finish association among source and goal. The conventions utilized at transport layer are User Datagram Protocol (UDP) and Transmission Control Protocol (TCP). IoT uses UDP as it is connection less protocol. So, the response time increases in IoT.

- **Application Layer**: The real use of brains of IoT is acknowledged at application layer. The application layer is used for social, industrial or personal purposes to retrieve the refined information.

The protocol stack just behaving like security component accessible on every layer is introduced in Table 1.2.

Table 1.2: Security Protocol Stack for IoT

Layer	Protocol Used	Security Protocol
Perception	IEEE 802.15.4 PHY, MAC	IEEE 802.15.4 security
Network	IPv6, RPL	IPSec
Transport	UDP	DTLS
Application	COAP	Not fixed designed by user

1.4 Applications of IoT

IoT can offer multiple applications and some of them are already in implementation. Miorandi in [19] forecasted that in coming future, IoT can be used for smart communications, intelligent transport, smart homes, intelligent business, and many more. The current applications of IoT with their change from exiting technology and advantage over it are presented through Table 1.3.

Table 1.3: Comparison of Applications based on existing and IoT Scenario

Application Area	Existing scenario	IoT Scenario	Advantage
Supply chain Information	This sector of supply chain information is usually handled manually. Offline registers are maintained to store the data about the products. This system is less flexible and don't have flexibility.	IoT can be used in any business related to supply chain model. It is capable of retrieving all the information regarding the things using the RFID mechanism.	Less man power required, increases efficiency, dynamic in nature as orders are maintained on the demand basis.
Health Related	Currently health of a patient can be monitored if he is physically present in the hospital and continuous manual monitoring of patients is done by the nursing staff. This is not efficient as it leads to wastage of time and is	In IOT scenario the mobile phones can be equipped with RFID or sensor equipment so as to continuously monitor the patient. This scenario will work efficiently in case of accident or in case if a patient is having	Efficient way of monitoring requires less man power, fast monitoring, remote monitoring.

	also a matter of concern if a patient is not handled at time of accident.	problem of high blood pressure, heart problem, or cancer.	
Smart Home	Now a day's almost everyone is having mobile phones but their technological aspect is missing. The only thing that is done to make our home smart is to have fire alarm facility. Existing technology like cell phone can be used to make home smart.	IoT make use of small sensors or RFID in things like refrigerator, lighting, living room for the real time information. If no one is in the room the lights should switch off automatically, a text message is sent to your phone if you forget to off AC, geyser.	Safety of homes. Better living, saves resources.
Insurance company	Insurance companies are giving same rate or premium to all the customers. Whereas some customers very often need this insurance claim. So the rate or premium to be charged is not flexible.	RFID can be used in human body or car of the customer to check heart rate, temperature, blood pressure. In car speed, acceleration, disturbance can be measured. Based on these parameters insurance officer can decide the rate and premium of customer.	Fast access of information, Benefit to insurance company as more customers will join them.
Agriculture	The present-day scenario of agriculture is that farmer cannot identify the disease of animals, breeds at early stage.	IoT will help the farmers to get the intact information about the breeds or animals every minute.	Better monitoring, direct communication between consumer and producer.

1.5 Challenges in IoT

Things in IoT connect through internet for taking effective decisions in real time. Gadgets in IoT are asset compelled in nature as far as their capacity, power, and transmission capacity [20].Many researchers are working on identifying the issues that need to be addressed in IoT environment for their efficient working. The major challenges faced in IoT are –

- **Heterogeneity:** The first major issue faced in IoT scenario is the interoperability of the heterogeneous devices. The hardware components as well as communication technology varies from one device to other. Thus a middleware is required to enable this communication.

- **Scalability:** Increase in number of connected devices will further make IoT vulnerable to new challenges like bandwidth issues, increased latency, and so on. An IoT network should be flexible enough to incorporate new things that get connected to it with time.
- **Reliability:** Devices in IoT communicate with other devices. Failure of any one could not affect the normal execution of the network.
- **Data storage and management:** Data from assorted sources is stored and maintained differently. Mining becomes another challenge when extracting information from diverse data.
- **Security:** All the above stated issues unleash some more issues in IoT like privacy and security. With more number of heterogeneous devices communicating and sharing the data, it becomes important to have secure framework for communication and at same time maintain the privacy of data for each device.

1.6 Security Requirements in IoT

Enormous information is collected in applications of IoT because of interrealtion of various items which calls for top of the line security [21]. IoT security spins basically around three viewpoints application security, system security, and network security [22]. Application security is utilized to take care of issues in applications like keen homes, urban areas, and so forth. System security keeps up security and protection of things the same number of the associated items regularly stays unattended. Network security arranged security chips away at offering security while correspondence through remote connections. With the expansion in web utilization, every client needs to get to any object of its decision that is associated with the system. This tremendous utilization of data and assets requires an exceptional consideration on security side of IoT.

Few security standards or rules need to be fulfilled in order to accomplish a safe as well as solid correspondence support to clients, things, and programming.

- **Confidentiality:** Source end to beneficiary end correspondence should be made safe. Information gathered using sensor is conveyed to beneficiary securely. For any case, it can't be guaranteed that sensor won't uncover this data to a unapproved object [23]. Full data executives and development procedure ought to likewise made safe [24].

- **Integrity:** During transmitting data from one end object to second, it ought not to be altered through intruder objects. Fundamental point of forcing integrity check is for privacy protection of client by avoiding the change of information.

- **Authentication:** This is utilized in order to start correspondence among objects after commonly recognizing one another [25]. Confirmation system diminishes the likelihood of attacks in the system. The significant test confronted while validation is because of distinguishing proof procedure of differing objects and recently associated items [26].

- **Availability:** As various objects are associated with IoT, idea is to guarantee availability. Availability isn't just for the information, it very well may be as far as administrations of applications, objects, and assets. Assured intrusion identification system may be utilized so as to identify the malicious or defectively activity carried on objects part of IoT.

- **Lightweight Solutions:** Objects bearing constrained resources shaping IoT need light operated security answers for execution. Lightweight arrangements don't plan to be feeble rather they involve less computational and control capacities for their preparing should be perfect in IoT. This is the reason lightweight arrangements act apt in task of IoT for secure usage and their long.

1.7 Security in IoT at Architecture Layers

Common engineering of web can't be straightforwardly connected in IoT scenario, as IoT expands correspondence among objects and human, and objects with different objects [27]. Establishment of Internet Engineering Task Force (IETF) and Institute of Electrical and Electronics Engineers (IEEE) posses characterized list of institutionalized correspondence just like security conventions for IoT [28].

IoT engineering involves 4 working layer-application, transport, network and perception layer. Security conventions utilized in web design can't be actualized in IoT because of compelled necessities like restricted power of equipment dependent IoT hubs, or sensor hubs. Secure oriented IoT design is built by imposing security system at every layer in IoT engineering. General development related to security covers data calculation security, data correspondence security, data gathering security, and security of the physically conveyed hubs present at application, transport, network and perception layer as represented in Figure 1.5.

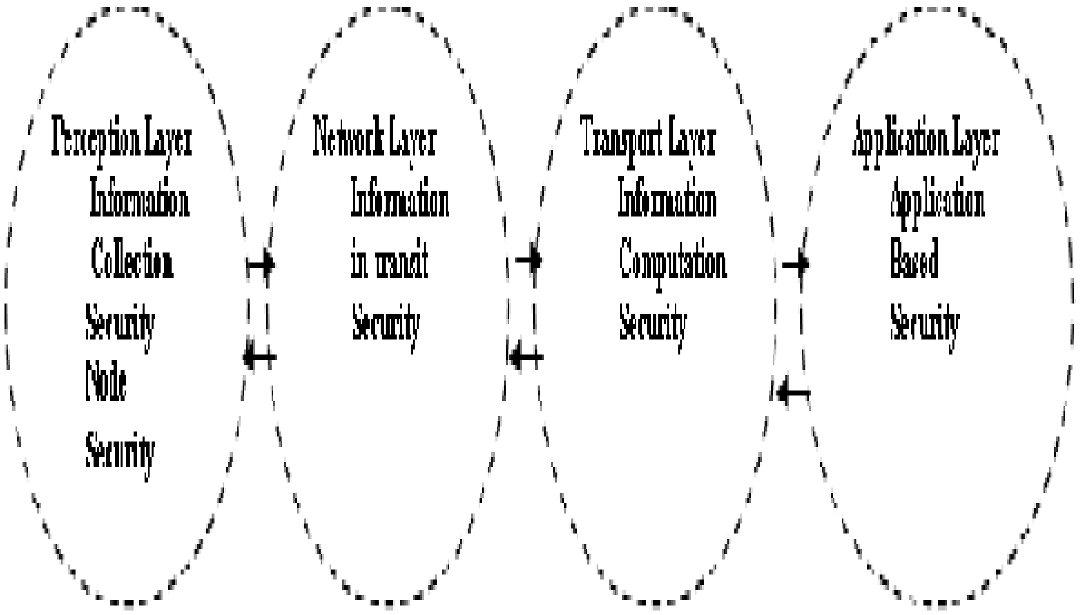


Figure 1.5: Security Construction Criteria at Each Layer

1.7.1 Perception Layer Security

This layer utilizes IEEE 802.15.4 standard by bolstering correspondence for small vitality gadgets. IEEE 802.15.4 chips away at both MAC and Physical layer with 250 bits/sec correspondence rate [29]. This standard proposes security on MAC layer which plays vital role in helping security systems for above layers in convention stack. The RF based handset created by Texas utilizes cc2420single chip. Crossbow manufactured TelosB bit gives security utilizing private key cryptography in IEEE

802.15.4 utilizing Advanced Encryption Standard (AES) [30]. Different methods of security for AES are utilized at MAC layer by IEEE 802.15.4 are indicated through Table 1.4. Distinctive methods of security in AES are arranged based on the security affirmations including degree of information coordinated.

Table 1.4: IEEE 802.15.4 Security Modes for AES

Security Mode	**Security provided**
No Security	Data is not encrypted
	Data authenticity is not validated
AES-CTR	Data is not encrypted
AES-CBC-MAC-32	Data is not encrypted
	Data authenticity using a 32 bit MIC
AES-CBC-MAC-64	Data is not encrypted
	Data authenticity using a 64 bit MIC
AES-CBC-MAC-128	Data is not encrypted
	Data authenticity using a 128 bit MIC
AES-CTR	Data is not encrypted
	Data authenticity is not validated
AES-CCM-64	Data is encrypted
	Data authenticity using a 64 bit MIC
AES-CCM-128	Data is encrypted
	Data authenticity using a 128 bit MIC

1.7.2 Network Layer Security

Sending IPv6 parcels over mixed systems requires a customary method. Subsequently, IPv6 Low Power Wireless Personal Area Network (6LoWPAN) is utilized with IEEE 802.15.4. This 6LoWPAN empowers start to finish correspondence utilizing IPv6 for obliged detecting articles in order to accomplish less power correspondence. 6LoWPAN can likewise work by utilizing low vitality of Bluetooth [31]. It don't carry its very own system of security rather it exploits AES component utilized on MAC layer for defeating weakness.

6LoWPAN utilizes Routing protocol for low power and Lossy systems (RPL) to adjust as per requirements of veer applications of IoT. Adjustment is done by RPL as per the requirements of a particular application. Necessities of applications are diverse for home applications, urban applications, or industry applications. One of three ways can be selected by RPL for tasks accessibility [32].

- **Unsecure Mode:** This mode is utilized by RPL as a matter of course for its task. No security component whenever forced on control messages is used for directing during this mode of un-security.

- **Authenticated Mode:** Utilization of the stated mode is done in case of objects going about like switches. At first, object may work like previously mentioned modes and utilizing accessible key however, later it may get another key using key sharing specialist to get approval and confirmation chasing.

- **Preinstalled Mode:** The officially accessible secret key is utilized by an object to link the current occasion in RPL. Object may go like a switch or host. Controlled directing messages offer integrity, privacy, and validation for this mode.

1.7.3 Transport Layer Security

Obliged conditions are imposed on IoT with applications using UDP at the transport layer. For safe correspondence at transport layer Datagram Transport Layer Protocol (DTLS) is favored over Transport Layer Security (TLS). Protocol DTLS keeps up an essential convention referred as Record layer, with other distinctive conventions at its best [33]. For applications of IoT, system for security is chosen relying upon correspondence essential highlights and dangers it may defeat [34]. Security insurances are given utilizing DTLS like secure correspondence setup, key administration as well as sharing. System is secured by allowing genuine clients to get into system. DTLS utilizes validation convention which introduces connection between new object and current object in system. On the off chance that handshake is fruitful, two gadgets can begin speaking with one another [34]. When the objects are verified, correspondence link is maintained where transmission of messages is done by scrambling it using one of a kind and fresh created key for session. DTLS likewise gives component of secure key administration and sharing between objects.

1.7.4 Application Layer Security

Obliged convention is utilized on the application layer of activity involving IoT. Messages at CoAP are verified imposing DTLS restriction for every message produced at individual layer. Hence, CoAP takes benefit of DTLS for offering security at application layer to the messages [35].

At transport layer, utilization of security instrument as DTLS mirrors that no primitive security system is accessible exclusively at this layer. DTLS operation includes similar strides where beginning one is process for handshake, secure set up for correspondence, and key administration/sharing. Leading to extra cost obliged IoT objects for handshake as well as extra overhead for every transmitted message. CoAP can likewise utilize its primitive security instrument alongside predefined security of DTLS. CoAP takes a shot at four distinct modes of security –No Sec, Pre Shared Key, Raw Public Key, and authentication mode. In this manner, both asymmetric and symmetric cryptographic system may be utilized with CoAP. CoAP's current execution is utilizing ECC or AES for its activity [36].

Every layer at IoT claims correspondence conventions along with security system accessible in each of layers. The inserted security arrangements at every layer posses specific issues that lead to prompt different attacks.

1.8 Attacks in IoT at Architecture Layers

Every IoT layer bolsters security components just referenced previously. Notwithstanding of previously mentioned accessible approaches of security, each layer is helpless to distinct type of attacks. With every different layer of operation in IoT, a unique attack can be implemented and is equally harmful to the users of the scenario. The stated area uncovers assault weakness at every IoT layer.

1.8.1 Perception Layer Based Attacks

Advancements, for example, Sensor based hubs or RFID are accessible at this layer that presents it powerless against below stated attacks:

- **Capture Attack:** All the nodes of system are specifically caught through attackers either manually or by adjusting product. Along these lines, the hubs don't fill in according to their normal usefulness.
- **Atmosphere Attacks:** Ordinary misfortune such as unpredictable breeze or snow can crush hubs utilized in detecting and further restrict their typical working.
- **Power Loss:** The sensor objects are control by battery that lessens after short span of time. Power misfortune may influence ordinary usefulness and tend to make circumstance for refusal of administration.

1.8.2 Network Layer Based Attacks

Network zone conveys gigantic measure of information from numerous gadgets. Major concerns like blockage of system or respectability in information can emerge at this layer. Attacks existing on this layer are stated below:

- **Denial of Service (DoS):** Invalidated objects may over-burden different objects or the server framework by generating counterfeit messages. Therefore, at last it cleave off the administrations of real objects to different objects requiring their administrations.

- **Privacy Disclosure:** Gathered data through various clients is put away on cloud or may be some other auxiliary mechanism of capacity. Individuals maintaining access of alteration to this put away information that may make unapproved disclosure to information and will escort the security hole of the clients.

1.8.3 Transport Layer Based Attacks

This layer works for information accumulation utilizing RFIDs or sensor motes. Accordingly, attacks might be infused at mote level that lead the interloper to veer off typical task by supplanting the first code. The attacks existing on transport layer are:

- **Eavesdropping:** Participation of web and remote correspondence mode can make objects powerless against listening in attack. Along these lines, different objects can constantly screen information of trade off object and can likewise send counterfeit messages to gather individual data of that object. This will lead to potential loss of information of a user.

- **Noisy Data:** The information is transmitted remotely with a chance that it is over vast separation, it might prompt loud expansion in it. Leading to influence the innovation of message as substance may be lost, or harmed. False information confusion may likewise is its serious outcome.

- **Sniffing:** Intruder objects are placed nearer to authentic sensor objects from the attacker so as to get data from genuine objects.

1.8.4 Application Layer Based Attacks

The application program bugs are launched by the attacker that may hinder the normal function of application.

- **Illegitimate Data Access:** IoT applications include various categories of clients. These clients effectively take an interest and may produce pernicious data. Consequently, solid confirmation instrument is necessitated that will keep up the security of system.

- **Software Bugs:** Different software engineers compose code that don't pursue standard example. This prompts cradle flood and obstruct the administrations to the authentic clients. Noxious clients assume control over this advantage to overrule the entire framework.

In view of the previously mentioned security methods selected by empowering advancements at every layer, specific powerless attacks can be distinguished on every layer and condensed in Table 1.5. Table 1.5 shows the used technology, method of security, and probable attack. This table will help to figure out the existing security mechanism operational on different layers of IoT and the corresponding attacks and the risks involved.

Table 1.5: Existing Layer Wise Security Mechanism and their Vulnerabilities

LAYER	TECHNOLOGY USED	SECURITY MECHANISM	ATTACKS
Perception Layer	RFID/ Sensors	IEEE 802.15.4 Security using AES modes	Atmosphere Attacks. Capture Attack, Power Loss
Network Layer	IPv6 Addressing, 6LoWPAN	Inbuilt RPL Security	Privacy Disclosure, DoS
Transport Layer	UDP	DTLS	Noisy Data, Eavesdropping, Sniffing
Application Layer	CoAP	Inbuilt DTLS, Cryptography solutions	Software Bugs , Illegitimate Data Access

1.9 Research Gaps

A detailed literature study was conducted on IoT. It has been observed that security is the prime concern in IoT these days. Therefore, it has resulted in drawing out the following research gaps:

- Lack of human intercession may prompt physical harm to the IoT objects.
- In built security solutions at transport layer using DTLS protocol is still vulnerable to attacks like eavesdropping and man-in-middle-attack.
- Devices in IoT are resource constrained in context of bandwidth and their power. Therefore, implementing complex heavyweight solutions for security may restrict the device efficiency.

1.10Problem Statement

The practical acceptability of IoT depends majorly o privacy and security of users. With the admittance of IoT in work areas, homes, business applications, or social locations lead to open vulnerabilities in privacy and security of IoT. This openness may inject serious attacks in IoT causing a measurable loss that is even difficult to imagine. Numerous attacks exist on IoT such as DoS, spoofing, eavesdropping, false injection of signal, replay attacks and many more [37]. Due to these numbers of possible attacks the security services offered by IoT such as authentication, integrity, and confidentiality are affected. Moreover, attacks will affect the users' privacy too. In [38] it is surveyed that IoT offers inherent primeval security solutions at each layer like CoAP at application layer, DTLS at transport layer, RPL at network layer which are still vulnerable to attacks. Therefore, the state-of-art carried on IoT lead to the identification of the following problem statements considered to carry out the research work.

Problem Statement 1: Different heterogeneous devices become a source of data collection in IoT. How data can be collected from different formats and mapped in one and how it is stored in IoT?

Problem Statement 2: Constrained operation of IoT devices due to lack of resources like power, the primitive authentication and cryptographic schemes are not optimum for IoT. Therefore, cryptographic solutions that are lightweight in their operation suit IoT scenario. Different symmetric as well as asymmetric lightweight solutions are available such as HIGHT, AES, RC5, RSA, PRESENT, ECC. These set of existing methods have more code length, execution time, and memory requirements and thus do not offer high security level in real time communication.

Problem Statement 3: How to diminish the execution time of the security protocol. The execution time of security solution is the time taken for key distribution and management, to encrypt and decrypt. The symmetric solutions take less execution time but can offer only integrity and confidentially but no authenticity. On the other hand, asymmetric solutions take more time to execute due to the large key size. Therefore, collection and computation of real time information is difficult and will take away IoT resources.

Problem Statement 4: What are the different performance parameters available to find the effectiveness of security mechanism?

Problem Statement 5: Is there any efficient way of mining the correct information from the data collected for fast decision making?

Therefore, a framework for IoT ensuring security is required that will ultimately offer authentication, confidentially, and integrity with secure key sharing method that will provide access to only legitimate users and will have less code length, least execution time and memory requirements. This work focuses on designing a hybrid lightweight logical security framework for IoT.

1.11 Motivation

Considering the quick moving time of connected objects, data gets traded among virtual and physical items consistently dedicating them as a piece in IoT. IoT includes different detecting objects for data gathering, for example, sensors, RFID, or Global position system (GPS) [39]. When information is gathered, this crude information gets changed over to important data that can be utilized for settling on choices progressively. IoT is developing as a scholarly innovation which is being utilized in different applications such as social insurance, urban areas, businesses, stock and so forth., for making them progressively effective and intuitive [40]. IoT incorporates different challenges such as bandwidth, power, scalability, security, heterogeneity, and privacy. Privacy and security is most crucial challenge to resolve in order to maintain users trust in IoT. Enormous objects interface with IoT through web that includes their own data subtleties also.

Consequently, to keep up individual data and security of every single user progresses toward becoming duty of IoT to keep up the survival of this innovation. Available solutions for security for every layer in IoT are still susceptible to attacks. So, cryptographic solutions are required for security assurance. The traditional algorithms are heavy in terms of their execution and do not fit in the IoT. Therefore, the algorithms that are lightweight are required for IoT.

Writing overview completed in most recent couple of years by numerous scientists proposed for IoT different lightweight symmetric and asymmetric security answers. Symmetric arrangements give secrecy, honesty, with little key size, and are not intricate but rather they don't offer legitimacy and dissemination of keys in them is a critical task. Then again, deviated calculations give privacy, uprightness, and legitimacy, yet the extent of key is too substantial making them complex and not fitting for obliged situations of IoT. In this way, there is a call for secure calculation which deliberately highlights of lightweight symmetric and asymmetric solutions to greater extent that may require less investment for execution along with ideal energy necessities and finally will guarantee full security administrations such as secrecy, trustworthiness, and realness.

1.12 Research Objectives

Looking at the increasing interest of users in smart devices to make smart things happen, the number of connections in IoT is increasing. This exposure of information may lead to the breach in security of connected devices and their owners. Therefore, in this research work the main objective is to propose a hybrid lightweight logical secure framework for IoT that will tend to improve security services like confidentially, integrity and authentication. The security framework will let only legitimate user to access the IoT services and will maintain the trust of users by securing data from unauthorized reading and modification, and preserving privacy of the users. This will automatically generate the interest of new users in IoT applications and will make our life manageable.

To achieve this goal the work is divided in multiple research objectives as stated below:

- To propose a new hybrid lightweight logical security framework to assure confidentiality and integrity of information with optimal use of resources.

- To compare proposed hybrid lightweight logical security framework with the existing frameworks in terms of their Key Size, Key Distribution Mechanism, Execution Time, Energy Efficiency, Memory Requirements.

- To validate the proposed framework towards various attacks and evaluate performance in terms of throughput, latency, and packet delivery ratio.

1.13 Research Methodology

Based on the problem identified from the literature survey conducted, the proposed methodology of the proposed research work is represented in Figure 1.6.

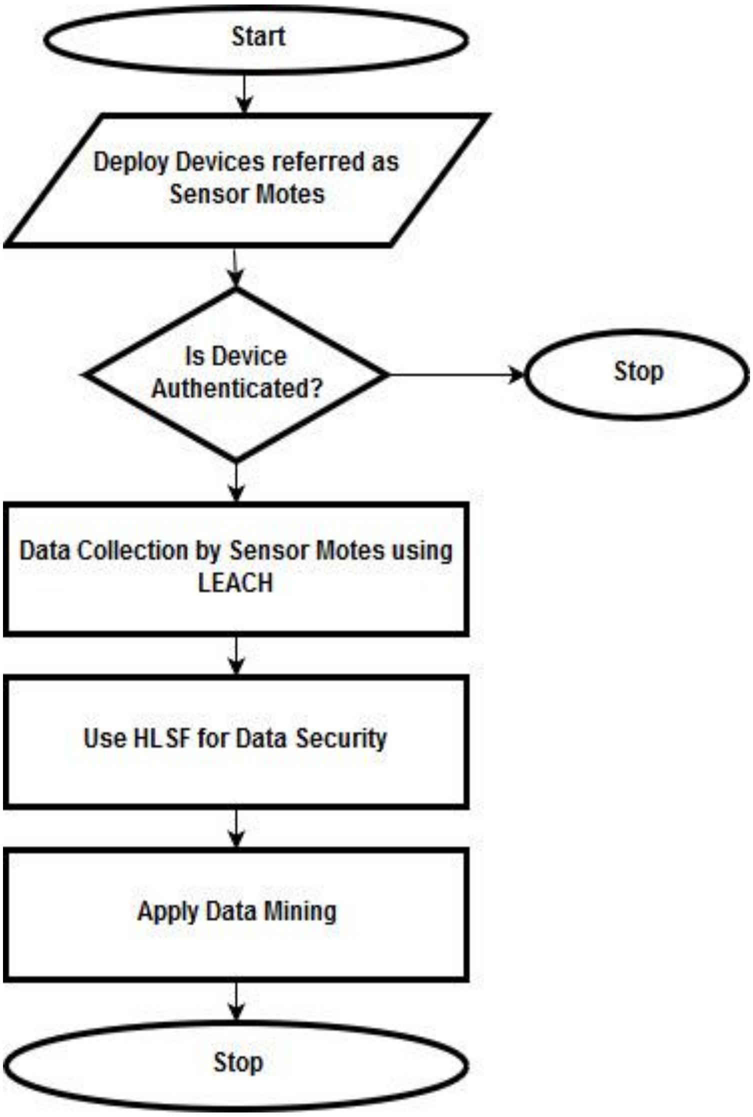
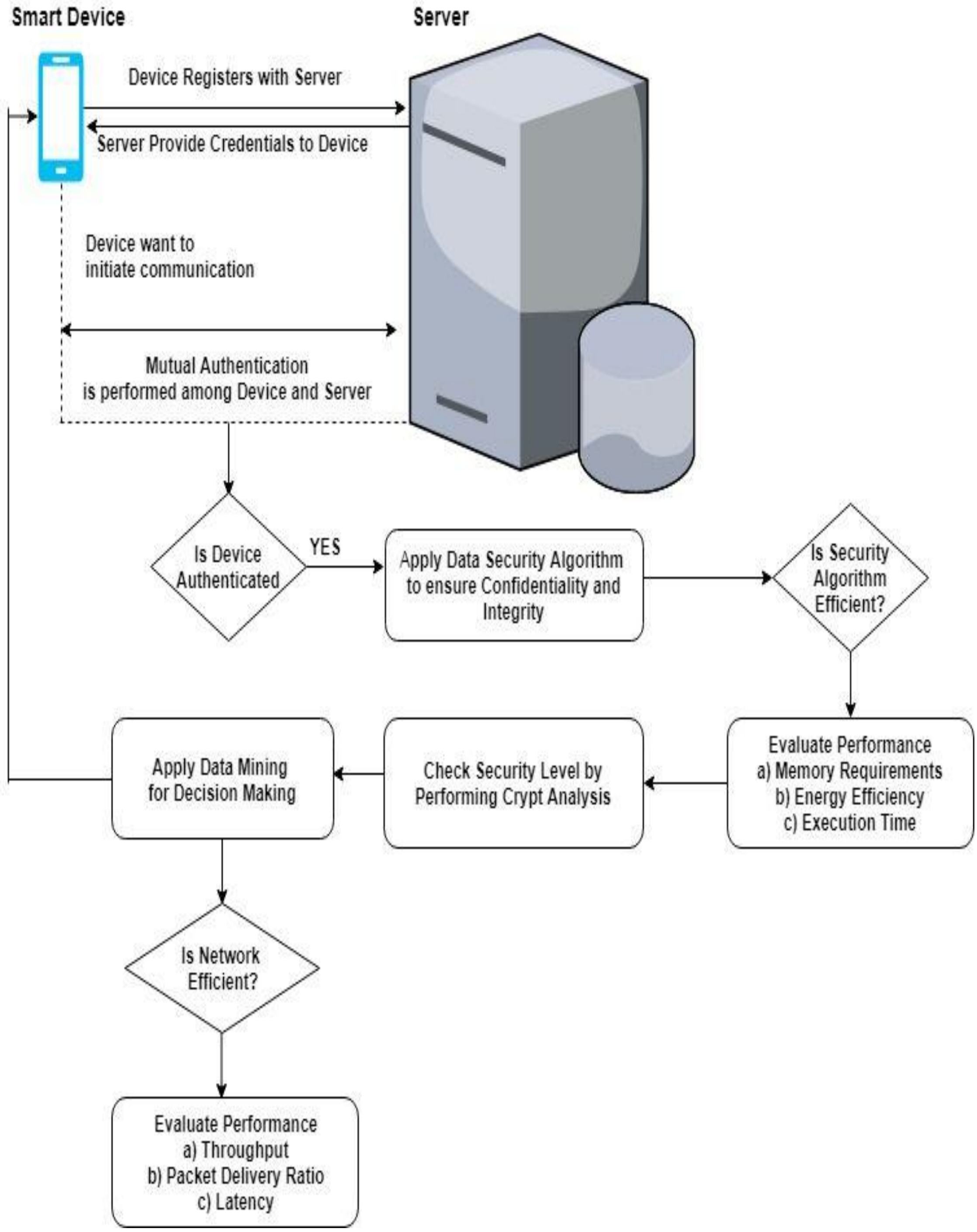

Start
Deploy Devices referred as Sensor Motes
Is Device Authenticated?
Stop
Data Collection by Sensor Motes using LEACH
Use HLSF for Data Security
Apply Data Mining
Stop

Figure 1.6: Flowchart Representing Research Work

The working model of the research work consists of smart devices embedded with sensor technology that wish to communicate. A centralized server is used to authenticate and validate each device that wish to become part of network. Figure 1.7 shows working process of a single device. Number of steps are involved when device wish to communicate with the server. First, device and server mutually authenticate each other. Later data confidentiality and integrity is implemented by security algorithms. Once any updated information is shared with the server, it maintains the record in database. Data mining is applied at server side to make meaningful decisions from the data collected. Finally, the decision is notified to the smart device, according to which devices proceeds further.



Smart Device
Server
Device Registers with Server
Server Provide Credentials to Device
Device want to initiate communication
Mutual Authentication is performed among Device and Server
Is Device Authenticated
YES
Apply Data Security Algorithm to ensure Confidentiality and Integrity
Is Security Algorithm Efficient?
Evaluate Performance a) Memory Requirements b) Energy Efficiency c) Execution Time
Check Security Level by Performing Crypt Analysis
Apply Data Mining for Decision Making
Is Network Efficient?
Evaluate Performance a) Throughput b) Packet Delivery Ratio c) Latency

Figure 1.7: Working Model for the Research Work

1.14 Research Assumptions

The work done in this research is based on certain research assumptions to carry out the study. Following are the research assumptions made before conducting the work:

- This work assumes to secure data communication among devices using a lightweight cryptographic solution.
- This work may be carried out on a working IoT scenario laid out in 1000*1000 square meter area.
- This work assumes the devices are statically deployed which preserve the consumption of power as mobility needs more power for operation. In IoT based scenario, efficient use of energy is the biggest challenge.
- The framework assumed for security will tend to offer authentication and confidentiality that will implicitly offer integrity, authorization by providing the lightweight solution for doing this.
- The devices in scenario will get registered and authenticated first time when they connect the network and data security is applied every time the data is collected and communicated over the network. On next restart, again fresh registration id and authentication process will follow.
- The packets are processed on the basis of first come first serve, no preemption is granted to any packet in any situation. All the packets are treated same.
- Existing solutions of security in IoT are vulnerable to known plaintext, chosen cipher text and chosen plaintext or differential attacks. If we try to mitigate the differential attack using algorithm it will also mitigate the other attacks in the system. Therefore, this research work improves the security solution by making it less vulnerable to differential attack.

Considering these assumptions for carrying out the research work, the solution is proposed that will offer various security services like authentication, confidentiality will maintaining the lightweight constraint on the devices. The framework will also mitigate the differential attack that will help to mitigate the further attacks. Finally the full framework will be tested by imposing mining and making decisions.

1.15 Major Contribution of Thesis

This research work is carried out to figure out an efficient method of offering security in IoT. For this a hybrid lightweight logical framework is designed that ensures authentication, confidentiality, integrity, and ultimately leads to efficient decision making. Initially the conceptual foundation is laid out by analyzing the existing work in literature and later the experimental analysis is made on the work carried out to prove the authenticity and the efficiency of the work done in this research.

1.15.1 Conceptual Foundation

- Existing security solutions in IoT are analyzed by determining their key size, energy efficiency, execution time, and memory requirements.
- Vulnerabilities of existing security solutions are identified.
- A lightweight framework is proposed to offer security in terms of authentication, confidentiality, integrity. The framework is lightweight as it requires less memory and time to execute and is more energy efficient.
- Frequent pattern mining approach is used for mining the data and hence taking the right decisions from data collected by the smart device.

1.15.2 Experimental Analysis

- Existing lightweight security algorithms are compared on the basis of their memory requirements, energy efficiency and execution time to find the better performer out of all.
- The selected existing algorithm is then compared with the proposed security algorithm.
- Cryptanalysis is performed on proposed security algorithm.
- Performance of Proposed hybrid lightweight logical security framework is compared with existing security frameworks CoAP and OSCAR in context of latency, throughput, and packet delivery ratio.

1.16 Organization of the thesis

Chapter 2 presents the review of literature that is required to know about the existing state of art for the chosen problem statement. This chapter first highlights the underlying architecture and security support in IoT. Existing cryptographic solutions are discussed and compared and it has been analyzed why they cannot be used in IoT. Later, existing lightweight cryptographic solutions are discussed in detail specifying differences among them.

In Chapter 3, a Hybrid Lightweight Security Framework (HLSF) is designed. This framework consists of 3 phases to offer a complete security solution to devices interacting in IoT. Phase 1 is general registration whenever a new device joins network, it has to register to the central server. Phase2 is authentication is required to achieve mutual authentication among device and server. Last Phase3, proposes a Lightweight Data Security (LDS) algorithm for data communicated over network in order to ensure confidentiality and integrity.

Once the HLSF is proposed, Chapter 4 compares the performance of existing lightweight security solutions with the proposed algorithm in phase 3 of HLSF. The performance is evaluated in terms of execution time, memory requirements, throughput, security impact, and energy efficiency.

Chapter 5 performs cryptanalysis of existing lightweight security algorithm SPECK with the proposed LDS on differential attack. As literature shows SPECK is vulnerable to differential attack. This chapter analysis the behavior of LDS over SPECK when crypt-analyzed to identify the attack and vulnerabilities

Once the data is collected using the proposed secure and efficient framework, Chapter 6 works on finding the best data mining mechanism. Data mining is used to find the meaningful information for decision making purpose. The decisions made are validated by calculating the precision, recall and accuracy of the result. Frequent path approach is used to make the decisions.

Last Chapter 7, concludes the entire research work done. This chapter also helps the novice by purposing the future aspects of the research.

CHAPTER 2

REVIEW OF LITERATURE

This section presents the current information related to IoT by conducting theoretical and methodological study. Firstly a detailed review is conducted on various issues in IoT while its integration with other devices or technologies. Secondly, a major issue, that is, security is identified and is described in detail. Security in IoT is studied at two levels. First all the existing authentication schemes are discussed. Later, the cryptographic solutions available in IoT to offer data confidentiality are examined. Finally, the whole literature review is synthesized into summary at the end.

2.1 Introduction

IoT has evolved as an innovation of next generation in this world of smart devices. IoT intends to provide services for data collection, data management, data and device security for application development. Things or devices in IoT communicate and compute to make our lives comfortable and safe. In inventory automation, real time check on items, their information management, status management, monitoring can be carried out using IoT. Various organizations and working groups are focused on IoT today like Samsung, apple, Thread group, etc. According to press release by Dr. Hong, Samsung have already launched the smart home kits which work on different modules using same application.

IoT fills in as an augmentation of web as it selects web associations with play out its tasks. Things in IoT are unique and could be in any way similar to human, pen, watch, creature, entry way. The things in this world of IoT convey through equipment objects like RFID, sensors, mobile phones.

RFID works as a better solution over existing 1D and 2D barcodes technologies for IoT as it allows the updating of data in the chip. RFID operates from very low frequency band of around 125 KHz to very high frequency band of 5.8 GHZ. It do not require any intervention of human and can be used to recognize the fast moving objects and in difficult environments also [41].

WSN technology is acting as one of the backbone technology operating in IoT scenario. WSN offers data collection using sensor nodes. These sensor nodes are small sized, battery operated and can be easily implemented in every device or component that need to be linked in IoT [42]. Cooperative data collection and analysis is made through sensor nodes deployed in a geographical region. Joint estimation of parameters works on this concept where an observation is made on the single variable through numerous sensors and the collected data is sent to the service center to make intelligent decisions over it.

The volume of information is expanding each day with the expansion in number of utilizations like content, interactive media, pictures, or recordings. Correspondence of such information among different objects requires a protected IoT engineering. IoT comprising of trillions of smart things embedded with RFIDs, sensors, battery, and computation power, leading to more security concerns in IoT as compared to internet. Creators in [43-44] introduced some security difficulties of IoT as the current conventions for web can't be specifically utilized for IoT because of forced requirements like power and continuous correspondence in IoT.

2.2 Underlying Technologies in IoT

With the innovation development in IoT, the quantities of things or items associated with it are likewise growing, accordingly the transmission capacity necessity of IoT increments. Frequency groups can be authorized or unlicensed. Application such as 2G or 3G, make use of paid license bands. For other specific applications like scientific, industrial or medical, frequency bands that are unlicensed referred as ISM bands are used. These bands available are 915 MHz, 433 MHz, and 2.4 GHZ. For communication that is Wi-Fi enabled in IoT, 2.4 GHz band is used [45].

The eventual fate of IoT is determined in four distinct stages by EPoSS. In 2010, utilized for retail and co-ordinations, 2010-2015, for associating objects, 2015-2020, incomplete astute items, after 2015, smart working articles. Most noticeable operational application regions of IoT are recognized for observing condition, medicinal services executives, computerized brilliant homes, industry, retail, inventory network the executives, coordination, horticulture, and some more. This vast range of applications offered by IoT makes it more popular.

For the total arrangement of IoT diverse empowering innovations such as sensor or RFID are required. As referenced in [46] RFID in IoT may be utilized for distinguishing things and to find the present condition of things progressively like the area. RFIDs are utilized in application such as retail the board, security, transport frameworks, or stock administration. RFID utilize radio waves in order to recognize things exceptionally using electronic scanner tags.

As depicted in [47], design of RFID has three parts- RFID tag/transponder, reader in RFID, and antennas in RFID that keep up information on microchip. The noteworthy parts of RFID will be RFID tag and RFID reader. Tag in RFID is joined to each single object which is dynamic in system. It involves a punched microchip with one of a kind character of a specific object. Reader in RFID is utilized to get the data from tag and transfer it to the application framework [48].

In [49], it is extracted that as there is growth in technology the numbers of devices attached to internet are increasing. Therefore, each device must have unique identifier by virtue of which communication can be established with that device. When the devices or things communicate in real time they form internet of things. The things ought to be lucid, effectively available, locatable, and tended to by utilizing any methods for correspondence like web, LAN, WAN, RFIDs, or sensor systems.

In [50], it is mentioned that RFID is used at perception layer to identify the objects uniquely with the specification of the RFID reader. RFID tags are embedded in each of things and a reader will update the information of a thing by reading its tag. The problem that is faced by IoT is standardization, as things in IoT are heterogeneous in nature so a standard reader is required for collecting the information.

In [51] author identified the security threats imposed due to use of RFID technology in IoT. DoS attack can be generated by the blocker tag. This tag will keep on responding to the reader affecting the operation of reader to communicate with other tags. Moreover, other privacy concern is the personal details and location privacy of the user. Personal information of user can be available to unauthorized things also.

In [52] some challenges in use of RFID technology are described. First challenge is collision while communication, when multiple signals are passed from one reader to multiple tags collision may occur. Second and most prominent challenge is security and privacy of tags. Tags maintain information but unprotected tags are susceptible to distinct category of attacks such as DoS, eavesdropping, and can also read personal information stored in tags.

In [53] it is referenced that another innovation that can be utilized in IoT that is sensor systems. WSN is utilized to detect and gather data identified with an action in genuine condition and that data is transferred to system for creating reactions. The regions where WSN is applicative are temperature control, remote detecting, dampness control, military any debacle the executives. Be that as it may, WSN works just to gather the information and can't process the information for official choice making. Thus, IoT makes use of this favorable position of WSN to gather information, and later impose preparing on collected information to make productive choices.

In [54] author proposed an adaptive QoS test bed for sensor network that will be able to cope with the dynamic network changes. This will mitigate the challenges like scalability and cost issues using Contiki OS.

Thus, IoT took the standpoint of WSN for gathering information, and later impose preparing on this information to make productive choices. IoT works over WSN for application areas stated above and extend these in more operational manner [55].

2.3 Data Collection in IoT

IoT effectively transforms the present situation of web in a vibrant working web. For achieving this, IoT oversees pervasiveness of distinct objects furnished using remote advances like Bluetooth, RFID, or by sensors [56][57]. User must run with a remote innovation relying upon the useful territory of IoT. In applications where the working territory in IoT is inside restricted space Bluetooth can be picked, for instance, in computerized overseeing of telephone calls [58][59]. RFIDs are selected for those applications where requirement is to keep up data for each object on the different tag of RFID. In this way, a reader of RFID is utilized to get data from the tag, for instance, in stock administration [60]. At last sensors based innovation is utilized in the greater part of the utilizations of IoT these days, for instance, in keen homes, shrewd urban communities, traffic the board and a lot more [61]. Sensors used may be movement sensors, nearness sensors, temperature sensors, and some more. As of now, Amazon's Alexa is propelled which utilizes the process of voice directions for the IoT objects associated with the web [62].

2.3.1 RFID for Data Collection

Internet is used to gather information, to surf, to communicate among things. Internet is further extended to IoT that include smart things that can make decisions and take appropriate actions based on this information collected or communicated. IoT usage is expanding at similar rate as internet. One of the applications of IoT can be in modernizing the traditional inventory management system. In inventory store, items are placed in proper fashion arranged on shelves [63]-[65]. Customers visiting inventory store pick the items from the shelves, they can place an item in wrong shelf. It leads to a lot of movement of items in inventory store and the exact remaining stock of a particular item cannot be calculated. Therefore, it will waste the human efforts in re-arranging the items on shelf and will also affect the stock management. This can result in either overstock and may also lead to outage of items.

2.3.2 WSN for Data collection

WSN may opt for different techniques for data collection. Same techniques are used in IoT when WSN is used as underlying technology. Nodes in WSN are referred either as source nodes, that usually collect the data. Second are the sink node or Base Station (BS) that gather the data from multiple sinks and maintains it for further processing [66]. The data collection in WSN can be categorized as tree based, cluster based, multi path, and hybrid techniques [67][68].

- **Tree Based Technique:** Collected data is maintained in the form of tree. BS/ Sink node is at the root whereas the source nodes are the leaves. Data collected at leaves is passed to the root nodes. Tree based techniques have numerous disadvantages. First, in case of packet failure of a particular source node, that data packet will not be available at the related sub tree at all. Second, the data packet sent from the source node at last level has to traverse every node in path till sink.

- **Cluster Based Technique:** Here whole area is divided into distinguished clusters, with a special node nominated as cluster head. All the source nodes in a particular cluster disperse data to the respective head of cluster that further dispense the data to sink node. Cluster based technique overcomes the disadvantage of tree based technique, by making direct connection between source and sink node. Therefore, if one link fails it is not going to affect the other links. Low Energy Adaptive Clustering Hierarchy (LEACH) protocol works on the clustering technique for data collection and is the basic protocol used in number of scenarios.

- **Multi Path Technique:** This technique uses the basic concept of tree based technique by overcoming its disadvantages. In this one source node can send the data to sink node through multiple paths instead of single path in tree based technique. Using this technique, system becomes more robust and if one path fails information can be still sustained from the other path.

- **Hybrid Technique:** This technique can be combination of tree, cluster or multipath depending on the application and the network performance required.

2.4 Security Insights of IoT

Considering the security aspect of IoT, security architectures, cryptographic algorithms are available in literature [69][70]. In [71][72], majorly three sides of security are referred by the researchers in IoT: Data security, Network security and System security.

- **Data Security** focuses on preventing the data in transit or static. Data security is essential to maintain the trust and relationship with the user. Data generated from different IoT based applications like smart grid, smart home should be kept safe from the disclosure to the outsider.

- **Network Security** considers the security of communication networks such as the technologies like RFID, WSN, or internet. The key generation algorithms, key distribution mechanism, secure cryptographic algorithms, access control mechanism need to be designing by focusing on the heterogeneity related to each connected device.

- **System Security** works on identifying the key design issues related to the complete IoT system. Various security solutions can be offered by security framework designed to accomplish the security and privacy challenges related to IoT.

In [73], it is referenced that IoT gangs a 3 layered engineering. The 3 layers in IoT incorporate perception layer at base, then network layer and finally application layer. Current design of web can't be utilized specifically for IoT. Because internet is used to connect various computers to form a network, whereas internet of things will consist of millions of connected devices which have communicating, computing and decision making capability. The features of IoT architecture are scalability, security, flexible, interoperability, and Quality of Service (QoS).

In [74], a 5 layered engineering has been proposed comprising of business layer, application, processing, transport, and perception. Perception layer operates in comparable approach to gather data at base dimension utilizing sensors or RFID. Transport layer is another identity of the network layer that is utilized to transfer data gathered from base layer. Next, Processing layer is included in design to recover, store and work on the information to change over it in important data. Application layer is utilized to fuse data handled cleverly in different applications such as industry, home, public activity, or military. Another business- the last layer is incorporated that goes about like general chief in IoT. Choice of which application and when will refer the data is taken by this layer.

In [75], an engineering that will keep up the security of information traded among the colleagues has been proposed. The amount of information is expanding each day from expansion in amount of uses such as content, mixed media, pictures, or recordings. Correspondence of such information among different objects requires a safe IoT design. To do this client confirmation, classification and honesty tests are connected to information. A progressively secure design is prepared to keep up IoT security.

In [76], an engineering where a settled confirmation conspires was utilized for personality of articles or things have been proposed. In [77], a security and quality mindful engineering that gives secure administrations to the things has been proposed, yet at the same time it don't defeats the open test to structure a system for speaking to IoT related information.

In [78], a architecture of IoT using embedded virtualization has been presented. This will encourage the communication among the heterogeneous devices in the IoT scenario. It makes use of the virtual machine based language, virtualization of the platform, cloud computations, and by virtual networks.

In [79], it has been described that IoT faces number of challenges like architecture as devices are heterogeneous in nature, scalability issues with each new device added to the network, real time computation, and search area. The searching in IoT by the sensors becomes more difficult due to these issues.

In [80][81], different challenges faced by IoT has been identified. One of the major challenges identified is the temporary data generated as well as redundancy of data is very high. Hence, data processing systems are required that will reduce redundancy and will convert data to make meaningful decisions. Other secondary challenges identified are scalability, and reliability on the data collected.

In [82], various challenges while integrating wireless sensor networks into IoT has been presented. The most prominent challenge is the security of sensor nodes, as nodes are accessed locally and remotely probability of attacks on sensor nodes is more. Other challenges are software and hardware related. Software related challenges include the type of data collection algorithm required for collecting the data so that energy requirement is less. Hardware related challenge works on reducing the cycle of sensor nodes in order to conserve energy and increase their battery life, so that they can work for the longer duration.

In [83], the security challenges in IoT are discussed. Security cracks can be due to hardware like sensors, RFIDs, software like communication and data collection protocols or due to middle ware that is internet. Security concerns of IoT are complex compared with web, as IoT comprises of millions of brilliant items with RFIDs, sensors, calculation control, and working on battery. These security risks can lead to various attacks on each layer of operation.

In [84], some security difficulties of IoT are referenced as the current conventions for web can't be straightforwardly utilized for IoT because of forced requirements like power and constant correspondence in IoT. The security challenges of IoT due to resource constrained devices are:

- Confidentiality: Just sender and receiver can peruse the information.
- Integrity: No middle man can alter the content the data while transmission.
- Availability: The services of the sender and receiver are always available.
- Authentication: Receivers should be able to verify the sender's identity.
- Authorization: Only valid users can access the data, unknown objects cannot access the data.

In [85], it has been mentioned that all services related to security should be incorporated in IoT in order to maintain trust of user in IoT. As IoT devices are resource constrained, therefore lightweight security solutions are required for their working, but still IoT is vulnerable to number of attacks. This work categories security attacks at different layers. Numbers of open problems in field of IoT are addressed like key management, identity management, access control, handling big data.

In [86][87], different attacks has been described that can be imposed on IoT. IoT offers primitive inbuilt solutions for security at each layer like CoAP at application layer, DTLS at transport layer, RPL at network layer that are still susceptible to attacks. The numerous attacks that can compromise the IoT scenario are summarized below:

- Availability of cloned objects by different manufacturer in markets.
- Masquerading attack
- Eavesdropping attack
- DoS
- Man in middle attack
- Differential attack
- Wormhole attack
- Saturation attack

Initial 2 attacks may be constrained considering articles from dependable producers. Alternate attacks can be dealt with by utilizing cryptographic arrangements. Number of security solutions are available in literature but the major concern is to evaluate the energy efficiency in offering the security for IoT.

In [88], an analysis is made on considering the efficient security solutions for IoT. The solutions mentioned are platform independent and helps in saving energy. Along with the energy efficiency, security of IoT is a vital issue in applications like smart healthcare. Guessing of password is the most ordinary thing that can happen in such applications employing a weak mechanism of security. Therefore, some password strengthening technique is used in [89] to maintain the privacy of user in the applications using IoT.

Security of IoT devices to ensure the validity of device and data collected from those devices becomes a vein in working of IoT. To ensure the validity of device various authentication schemes are available in literature that will help in validating the data collected from those devices.

The data can be further collected in secure way to enhance the trust of the third party in the IoT system. In [90], a sensing scheme that is based on policy and trustworthiness of the sensed data is proposed. The scheme proposed, Real Alert proves the trust of both the devices and the data collected from those devices. That will increase the belief of user.

The belief value that is observed by any node N during an event e is calculated as follows in equation 1:

$$belief_N(e) = \sum_{e_i \in e} M_N(e_i) \quad (1)$$

Here e_i represents the basic events that finally form the event e. M_N stands for the event e view. Therefore, the packet dropping level of a node Ni is represented by PD_{Ni} and is calculated as in equation 2.

$$PD_{Ni} = \sum_{s=1}^{S} M_{Ni}\,(N_s) \qquad (2)$$

Here $M_{Ni}(N_s)$ represents the node N_i view on node N_s. Precision (Pr) and Recall (Re) parameters are used for finding the accuracy of RealAlert as shown in equation 3 and 4 respectively.

$$Pr = \frac{Truenumberofcaughtmaliciousdevices}{Totaluntrustworthycaughtdevices} (3)$$

$$Re = \frac{Truenumberofcaughtmaliciousdevices}{Totaltrulydevicesthataremalicious} (4)$$

Constrained IoT works in real time to collect the data from the multiple devices. Therefore, an authentication mechanism that is location based is required to offer accuracy in the data collection [91].

Author in [92], have proposed a protocol (AAoT) for attestation and authentication of devices in IoT. This scheme does not require any changes in micro controlling units that are existing. The drawback of this scheme is that it cannot be patched for dynamic vulnerabilities. A scheme referred as Authenticated key exchange has been proposed in [93] that offers resistance alongside the side channel attacks and is flexible in the management of key certificates but is still vulnerable to leakage of random secret values.

In the current operations of IoT, new threats are being injected at every phase. Therefore in [94], a model used for detecting the threats has been proposed for

SCADA. Detection models using support vector machine and deep belief network are used for attack detection.

2.4.1 Conventional Cryptographic Algorithms

Cryptographic algorithms are used to work on the scrambled data to ensure it is non readable by the unauthorized users. It comprises of two phases one is encryption and the other one is decryption. Encryption is used to convert the original text into the jumbled code; on the other hand, decryption again converts the jumbled code back to the original readable text. The conventional cryptographic algorithms used are of two types- symmetric and asymmetric. In symmetric algorithms a unique key is communal among the sender and the receiver. Sender encodes the message using secret key resulting in a message that is not human readable. Receiver, on the other side, decrypts the message using same secret key and finds the original text out of it [95]. The phases of symmetric and asymmetric algorithm representing how they work are represented in Figure 2.1 and Figure 2.2 respectively

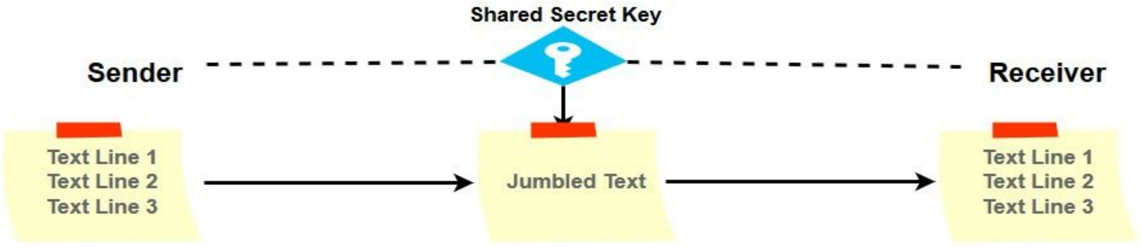


Figure 2.1: Symmetric Encryption

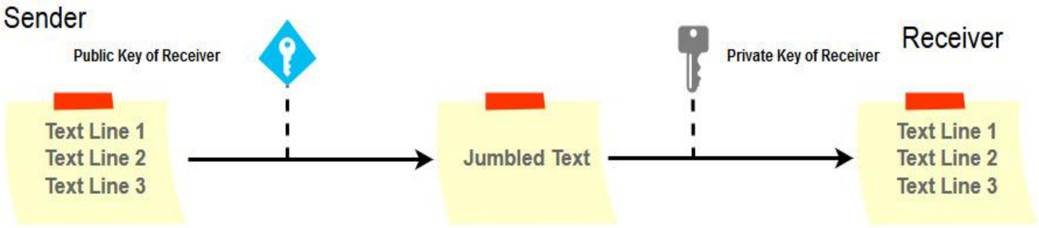


Figure 2.2: Asymmetric Encryption

Cryptography algorithms from last many years are working as key helpers for offering security services like authentication, integrity, confidentiality, validation, and non-repudiation. Generally, cryptographic algorithms are categorized into symmetric and asymmetric algorithms. Different algorithms in each category have been devised in literature.

In [96], conventional symmetric key algorithms has been surveyed, they can be substitution or permutation based. Substitution based will replace the plain text with new message referring some formula. Permutation based is used to transform plain text in a manner that it becomes difficult to attack by an intruder. Further symmetric algorithms may be classified as stream ciphers and block ciphers. First, stream ciphers work on one bit whereas block cipher works on entire block at a time. Various symmetric key algorithms exist like Data Encryption Standard (DES), Triple Des, Advanced Encryption Standard (AES), IDEA, Blowfish.

Customary Symmetric calculations DES, Triple DES, AES, Blowfish, IDEA are looked at based on their properties like data size, key size, number of rounds, structure and existing attacks (Table 2.1).

Table 2.1: Comparison of Existing Symmetric Cryptographic Algorithms

Algorithm	Data Size	Key Size	No of Rounds	Structure	Possible Attacks
AES	128 Bits	128/192/256	10/12/14	Feistel	Related Key/ Man-in-middle attack
DES	64 Bits	56	16	Feistel	Brute force
Triple DES	64 Bits	168	48	Feistel	Meet in middle
Blowfish	64 Bits	128-448	16	Feistel	Second order differential
IDEA	64 Bits	128	8	Substitution-Permutation	Related key
TEA	64 Bits	128	64	Feistel	Related key

Conventional asymmetric Algorithms can be used for Key generation, Electronic/Digital signature, Encryption. The most often used algorithms by RSA-Rivest, Shamir & Adleman are, Elliptic curve cryptography (ECC), Deffie helmen key exchange (DH), and Hash functions. All these algorithms have large key size as compared to symmetric algorithms, so the chances of attacks are very less. Whereas, due to large key size there implementation becomes very slow. In [97] two asymmetric algorithms used in IoT has been proposed

- **RSA:** It was designed by Ron Rivest, Adi Shamir and Leonard Adleman in 1978. RSA chips away at creating private and public key pair by choosing two expansive prime numbers. Discover their functions and picking aimlessly their key for encryption and in this way computing the unscrambling key. In [98], RSA is executed where open key is distributed transparently though private key is made secure. In [99], an increasingly secure RSA encryption has been recommended that is utilized to scramble and unscramble records for keeping up protection of client.
- **Elliptic Curve cryptography (ECC)**: This requires less bits in key when contrasted with RSA. Consequently having quick preparing and less prerequisites. It was designed by Victor Miller and Neal Koblitz in 1985. It is based on logarithmic framework by taking two points on elliptic curve. Discrete logarithm is utilized to create key that is utilized to process key. In [100], a safe execution on ECC for little regions is displayed that will prompt quicker calculations continuously. In [101], creator perplexed multiplication at ECC enhancement for 6LoWPAN hubs. As opposed to utilizing multiplication for augmentation, bit moving is utilized to enhance the utilization for low power gadgets. Author in [102] have proposed a curve based on elliptic curve cryptography that works for IoT applications in an efficient way by offering high speed and using less number of resources. As it offers a high security level it can be used further in protocols for cryptography. However, the algorithms like ECC can be easily broken using the quantum computing. In [103], various cryptography solutions are introduced that are resistant to quantum computers and work well in IoT.

Customary Symmetric and Asymmetric calculations are unable for IoT condition with constrained power gadgets, low computation, and few memory limit. Along these lines, lightweight security calculations were proposed in IoT. These arrangements are light as far as their memory necessities, key size, and execution time with the goal that less assets are used when contrasted with substantial weight arrangements.

Out of traditional cryptographic solutions, research has been carried out in literature by using AES as inbuilt security solution of CoAP at the application layer, but still the scenario is vulnerable to attacks like related key or man in middle attack. A plenty of lightweight solutions were proposed in literature that tend to work well in IoT and are described in next section.

2.4.2 Lightweight Cryptographic Algorithms

Lightweight refers to the algorithms that require fewer resources with optimal efficiency. The term lightweight does not refer that algorithm are weak in their operation. With the change in trend of upcoming applications using constrained devices, a lot of efforts had been made to optimize AES for these applications. But the adaptations made in AES were not appropriate as per the requirement of these devices. Although implementation of AES was made fast but still it is very complex and have large code that does not suites the requirements. Moreover, in some application scenarios such a heavy security like AES-128 offers is not required, making this block size of 128 as not optimum for use. Therefore a lot of work was focused to extend the vision on lightweight cryptography that can offer a good level of security along with less complexity and suites constrained environments. In 2013, SIMON and SPECK were commenced for the very first time by a researcher group in US National Security Agency.

The data security in IoT applications can be offered using symmetric, asymmetric or has functions. Symmetric algorithms require the processors with minimum cost indulged whereas asymmetric requires hardware configuration to be implementation due to complexity in their operation [104]. Authors have also evaluated the statistical error. For this measured results (M_i) over the total number of runs (R), the mean value $\overline{(X)}$, standard mean error $\sigma_{\overline{X}}$, and standard deviation (σ_X)are evaluated as in equation 5, 6 and 7.

$$\overline{X} = \frac{1}{R}\sum_{i=1}^{R} M_i \qquad (5)$$

$$\sigma_X = \sqrt{\frac{1}{R}\sum_{i=1}^{R} (M_i - \overline{X})^2} \qquad (6)$$

$$\sigma_{\overline{X}} = \frac{1}{\sqrt{R}}\sigma_X \qquad (7)$$

The existing lightweight algorithms are detailed below.

- **Advanced Encryption Standard (AES):** AES is utilized as an inbuilt arrangement in COAP at application layer. It is a symmetric square figure institutionalized by NIST. It utilizes substitution stage system and deals with 4*4 network having square length of 128 bits. Each byte gets influenced by sub bytes, shift rows, Mixed Columns, Add Round Key. Key size than can be utilized is 128, 192, 256 bits. AES is utilized as validation component in RFID based frameworks [105]. AES is as yet defenseless against man-in-center assault [106].
- **High Security and Lightweight (HIGHT):** In [107], HIGHT, a high security and lightweight cryptography calculation has been proposed. HIGHT utilizes extremely fundamental activities like expansion mod 28 or XOR to work for Feistel arrange. It has a square size of 64 bits, work in 32 rounds on128 bit keys. Its keys are created while encryption and decoding stage. In [108], a parallel execution of HIGHT that requires less power, referenced in few lines of code, and enhances speed for RFID frameworks has been displayed. HIGHT is powerless against immersion assault.
- **Tiny Encryption Algorithm (TEA):** In [109], a Tiny Encryption Algorithm is recommended that is utilized for compelled situations like sensor systems or savvy things. It is written in not many lines of code. It doesn't utilize an intricate program yet requires straightforward tasks of XOR, including and moving. It utilizes a square size of 64 bits and 128 piece keys and does not make utilization of existing tables or any predefined calculations. Number of variations exists for TEA like expanded TEA, Block TEA, etc. These expansions endeavor to determine the issues in unique TEA like proportionate keys. Because of its basic tasks TEA and its variation are defenseless to number of assaults like in man-in-center assaults [110].
- **PRESENT:** In [111], creator proposed PRESENT that depends on SPN and is utilized as ultra lightweight calculation for security. It chips away at

substitution layer utilizes 4-bit info and yield S-boxes for equipment advancement. It has key size of 80 or 128 bits and works on 64-bit squares. PRESENT has been exhibited as a lightweight cryptography arrangement in ISO/IEC 29192-2:2012 "Lightweight Cryptography". PRESENT is defenseless against differential assault on 26 out of the 31 rounds [112].

- **RC5:** In [113], creator initially authored RC5 for turns that are information autonomous. It gangs Feistel structure and can function admirably as lightweight calculation as it is utilized in remote sensor situations. RC5 is considered as w/r/b, where w alludes to word measure, r represents number of rounds and b will tell about the quantity of bytes in encryption key. RC5 by and large deals with 32 bit size however its variations can be 16, 32, 64. It can work for 0, 1,..,255 rounds utilizing 0,1,..255 key bytes. Standard key size is 16 byte on 20 rounds of activity. RC5 is defenseless against differential assault [114].
- **SIMON:** SIMON 2n refers to a block cipher having n-bit word forming a block of 2n-bit. The value of n can be 16, 24, 32, 48, and 64. SIMON 2n using key as k-word key (kn-bit) is referred as SIMON 2n/kn. Therefore, SIMON 96/144 will be working on a block of 96-bit plaintext and using key of 144-bits [115]. SIMON belongs to the family of block ciphers with varying block sizes. It can support 32, 48, 64, 96 and 128 bits of block size that further can work on varying key sizes as shown in Table 2.2.

Table 2.2: Block and Key Size Supported by SIMON

Block Size	Key Size
32	64
48	72,96
64	96, 128
96	96, 144
128	128, 192, 256

- **SPECK:** The specifications for SPECK are similar to SIMON. Therefore, SPECK 128/128 refers to SPECK block cipher operating on 128 bit block size suing 128-bit key. The block and key size supported by SPECK are same as of SIMON represented in Table 2.2 above [115]. SPECK uses feistel structure performing bitwise XOR, circular shits and modular addition in each round at both directions [116].

- **TWINE:** TWINE is a 64 bit block cipher forming a basic Feistel structure. Feistel functions consist of 16 4-bit sub blocks using key addition. Two key sizes 80 and 128 bits are supported by TWINE [117]. TWINE operates on total 36 rounds with same round function.

.

- **FANTOMAS:** FANTOMAS is an instance of LS-designs (LS consists of L-boxes using look up tables and S-boxes that are bit sliced). The FANTOMAS block cipher can be represented as s*L bits matrix [118]. Every block will represent one bit as shown in Figure 2.3 below

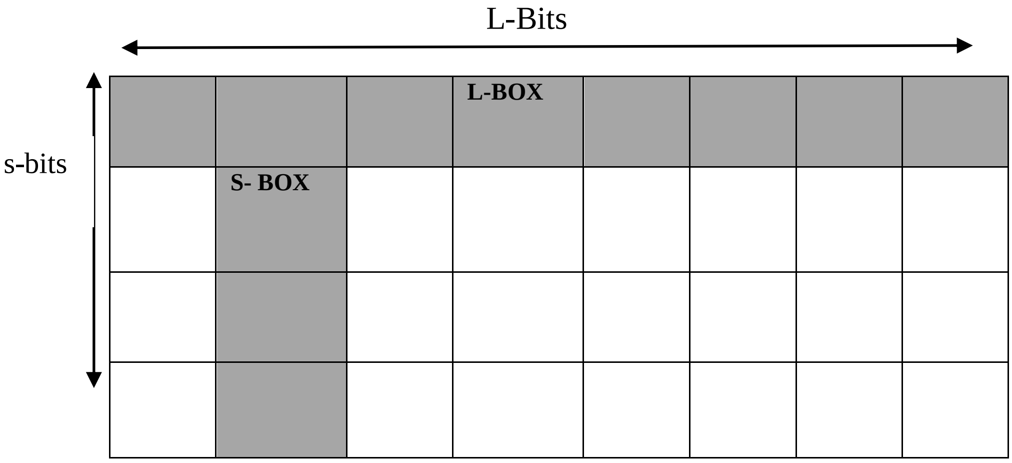


Figure 2.3: FANTOMAS Block Cipher

On each column of matrix, s*s bits permutation is applied, whereas on each line of matrix L*L permutation is applied. For example, considering 128-bit as key and block length of FANTOMAS. The s-bits will be 8 and L bits will be 16.

Comparison of all aforementioned Symmetric lightweight algorithms is made in table 2.3 based on their physical structure, length of code, rounds in total for their working, size of key and size of processing block, and at last the vulnerability of every algorithm is found for various attacks.

Table 2.3: Comparison of Lightweight Symmetric Algorithms

Symmetric Algorithm	Code length	Structure	Number of rounds	Key Size	Block Size	Possible Attacks
AES	2606	SPN	10	128	128	Meet-in-middle attack
HIGHT	5672	Feistel	32	128	64	Saturation attack
TEA	1140	Feistel	32	128	64	Related Key Attack
PRESENT	936	SPN	32	80	64	Differential attack
RC5	Not fixed	Feistel	20	16	32	Differential attack
SIMON	2200	Feistel	42	96	64	Meet in middle and differential attack
SPECK	1630	Feistel	26	96	64	Differential attack
FANTOMAS	5600	LS Design	16	128	128	Cache Timing Attack [119]
TWINE	4203	Feistel	36	80	64	Linear attack with zero co-relation

2.4.3 Attacks on Existing Algorithms

Existing security arrangements in IoT are as yet powerless against following attacks:

- **DoS:** It will stop the administrations of system for the approved clients because of access of system association demands from unapproved clients.
- **Eavesdropping:** Intruder can listen the correspondence among sender and collector. So this is attack on secrecy.
- **Man-in-Middle:** In this a delegate client can get the key of one of the sides and will begin correspondence as though it is the legitimate party.
- **Saturation:** In this intruder will endeavor to utilize the physical and mental capacity of approved party by its huge use.
- **Masquerading:** An interloper have the character of some other approved client. So it can tear down the assets of IoT.
- **Differential:** Change in input information will influence the output. So this attack can locate the key from system changes.

2.5 Data Mining in IoT

IoT effectively transforms the present situation of web into self-motivated working web. For getting over this limit, IoT oversees pervasiveness of distinguished gadgets furnished using remote innovations like RFID, Bluetooth, or by sensors. IoT plans to diminish the human intercessions and work adroitly dependent on detected data. The data gathered by sensors or readers is kept up on server which is sent for preparing to remove the profitable data for basic leadership. Satisfying the errand of data extricating, diverse Machine Learning (ML) techniques are accessible [120].

ML is utilized for preparing framework depending on expertise or tests accessible from history to own a closing expression related to specific issue. Distinctive strategies for ML are accessible for various fields like computerized reasoning, information examination, design revamping, DM and a lot more [121]. The ebb and flow inquire about IoT is centered around utilization of DM methods in relation to separate significant data from any occasion and after that take choices likewise [122].

For instance, IoT can be utilized in stock administration for monitoring the things at various occasions of a year. In view of the past information, a DM method may utilize to settle on choice that closeout of a specific thing like state, cold beverage, is high in summer when contrasted with winters. This entire system happens in a stock utilization of IoT including the period of nonstop detecting, information gathering, DM is spoken to through Figure 2.4 underneath.

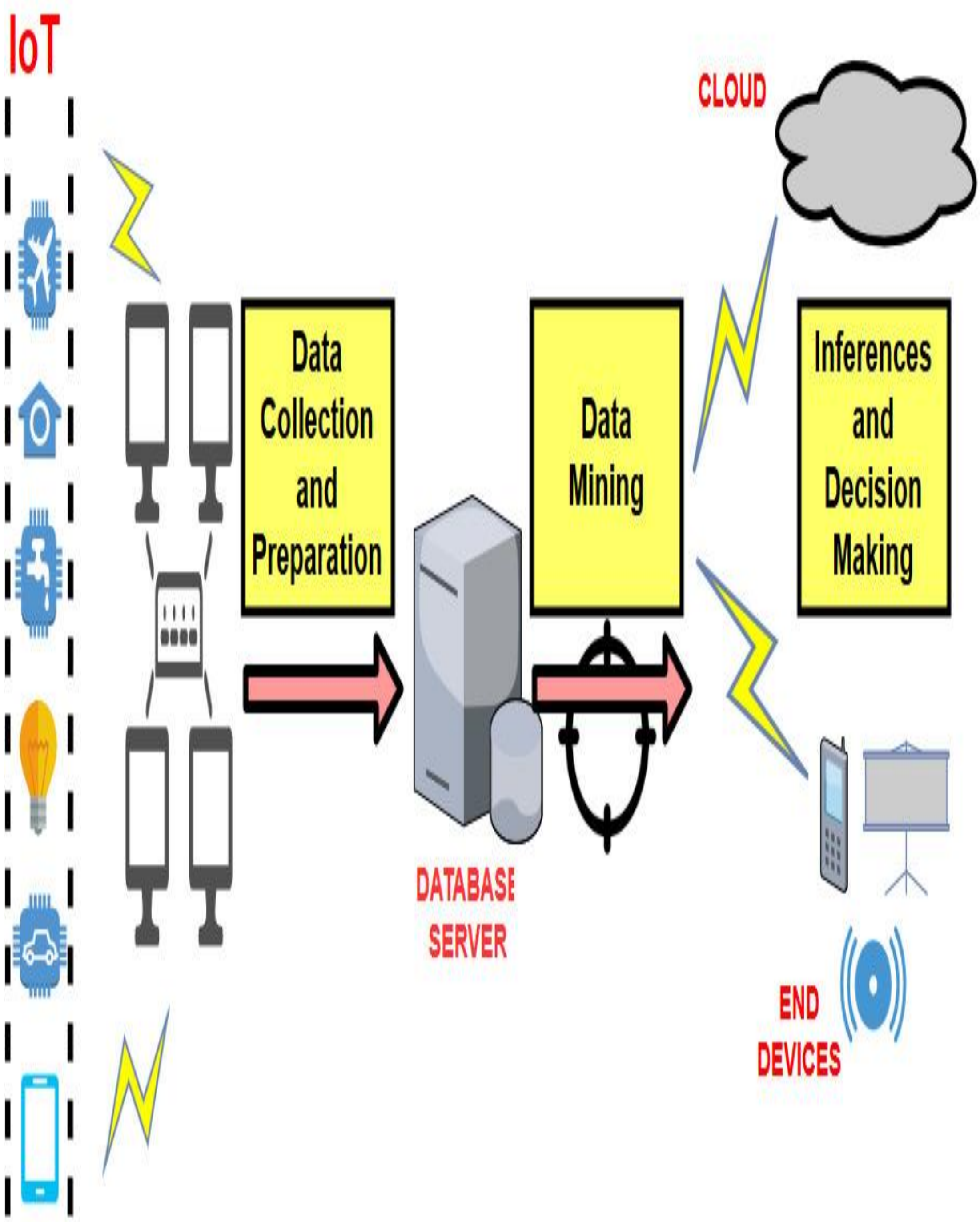

IoT
CLOUD
Data Collection and Preparation
Data Mining
Inferences and Decision Making
DATABASE SERVER
END DEVICES

Figure 2.4: Decision Making IoT Framework using Data Mining

2.5.1 Challenges using DM in IoT

In spite of the fact that DM is incorporated in applications based on IoT in same path such as utilized in different territories, yet at the same time it accompanies certain difficulties with IoT due to beneath expressed reasons [123][124]:

- **Large Repository of Data Collected**: With number of gadgets utilized for IoT task, every gadget produces greater part of information bringing about nonstop information of gigantic volume.
- **Lack Security Assurance:** Different new businesses having IoT arrangements cannot concentrate on security. This may lead to accumulation of gigantic information taken from even the un-believed sources squandering time and thus bringing down proficiency of DM method.
- **Heterogeneity in Data:** Distributed gadgets in IoT gather data from various wellsprings of shifting organizations bringing about information sources making the errand for DM testing.
- **Devices with Constrained Resources**: Expecting superiority from compelled gadgets regarding force or limit is required in IoT. Thus, ceasing IoT from requesting the standard coordination of security and prompting age of ill-conceived information.
- **Operation in Real Time:** IoT gadgets gather data at a particular schedule opening and will in general work progressively. In this way, time stamps should be recorded and choices ought to be produced utilizing DM procedure continuously considering it a test for DM strategy.
- **Information related to Noise:** IoT services gather little data that outcome in clamor or mistake during transmission or accumulation.

Thinking about every one of these difficulties, this examination presents diverse DM systems accessible for IoT applications. Right off the bat, the distinctive DM systems accessible in writing are talked about. Next, points of interest and drawbacks of utilizing a DM system for use in IoT are portrayed. In last area, an execution investigation is made on premise of exactness, review, and precision.

2.5.2 Existing Techniques for Mining Data in IoT

IoT offers opportunity for various gadgets to correspondence between themselves and furthermore to gather choices. Applications including IoT produce enormous measure of information using the sensors utilized for gathering the information. For winning these services some significant data should be found from the gathered information to decide. In this way DM method comprise of comprehensively three stages; initial step is readiness information where the information is coordinated from heterogeneous assets, clamor impact is killed and afterward the information gathered is transited to the subsequent stage of mining.

Next step emphasize on applying a DM strategy for discovering conduct of gathered information and take deductions lastly information is spoken to through services to the client. Distinctive DM procedures exist in writing however the stated exploration work extensively centers around three strategies to be specific clustering, classification, and association rule mining.

2.5.2.1 Classification

In order, there are predefined gatherings of classes that refers, the information is arranged dependent on past learning. Every class groups a different usefulness recognizing from different classes. At whatever point another information device is recognized, it is doled out to nay of the pre-set up class by knowing its usefulness. Hence, the fundamental point of order is to assemble the information in various classes. For instance, if the measure of motion picture ticket paid is less, client will sit in silver, sum is more he will get seat in gold and if the sum paid is most extreme the client will get seated in business class, this all will prompt grouping of the clients. Order can chip away at set apart just as plain information. The information that is checked can be utilized by the classifier as a capacity for future forecast. Then again the information that is plain will be characterized by the classifier.

For quantifying the adequacy of the order system, the quantity of information questions that may be distributed to correct class that is recognized. Along these lines, the Effective Rate (ER) of arrangement is resolved utilizing:

$$\mathrm{ER} = \frac{\mathrm{NR}}{\mathrm{NT}}$$

Where NR represents the number of data objects that are allocated correctly and NT represents the total data objects existing. Classification works better for a application if the ER value is high and in case ER is low, same application will fail for the classification. Numerous methods are offered in classification such as frame based expert systems, rule based, Bayesian network, decision tree, neural networks, and even more.

2.5.2.2 Clustering

Clustering isolates the plain information objects into significant gatherings with the end goal that the information protests that have a place with same gathering forces comparative qualities and the information questions that have a place with other gathering groups diverse attributes in the individual sense. For finding the proper bunch makes utilization of unsupervised learning. For instance, for a stock framework, so as to comprehend the buy conduct of the clients, bunching can be utilized to aggregate the clients with same buy propensities in a single gathering [125].

Along these lines, if a lot of contribution of plain items is given as I= {I1, I2, … .,In}, the yield will include separating input I in n bunches C= {C1, C2,… … ., Cn}, contingent on the qualities of each gathering. Diverse techniques are accessible to configuration groups, for example, k-implies clustering, progressive clustering, dividing strategy and some more.

2.5.2.3 Association

This way of mining is centered around finding the intriguing connections among the gigantic arrangement of gathered information. In this way, association mining is worried on discover the informative things that regularly happen together [126]. For instance, in stock administration, in the event that a client is acquiring a pencil, at that point he will purchase an rubber. Two components are utilized to quantify the connection, first is support and the other is confidence. Support decides the level of clients going for this buy. The help of 20% shows that among the considerable number of clients 20% clients will go for this strategy. Confidence decides the level of clients with precisely the same relationship alternatives picked for procurement.

Presently, the confidence of suppose state 80% shows that odds of rubber that will be picked with the pencil is 80%. For instance, if P signifies the buy of pencil and E indicates the buy of eraser. In this manner,

$$Support(P => E) = P(P \cup E)$$
$$Confidence(P => E) = P(P|E)$$

From the above talked about DM systems in IoT, their qualities, alongside the applications are recognized relying upon their task spoke to in Table 2.4 underneath.

Table 2.4: Distinguished Characteristics of Different Data Mining Techniques

Characteristic/ Application		**Clustering**	**Classification**	**Association Based**
Learning	Supervised		✓	✓
	Unsupervised	✓		✓
Previous Data	Maintained			✓
	Not Maintained	✓	✓	
Application	Industry	✓		✓
	e-commerce	✓	✓	✓

2.5.3 Arrangement of Data Mining Techniques

The above introduced mining strategies can be connected exclusively for IoT to remove information and continue for basic leadership. Be that as it may, the best way is to impose these mining procedures in mix so as to utilize the effective among all highlights relating to a mining system. There are 12 blends conceivable from these three methods. For instance, clustering procedure can be consolidated exclusively either with association based or classification mining or it very well may be joined with two other methods in various request as appeared in Figure 2.5.

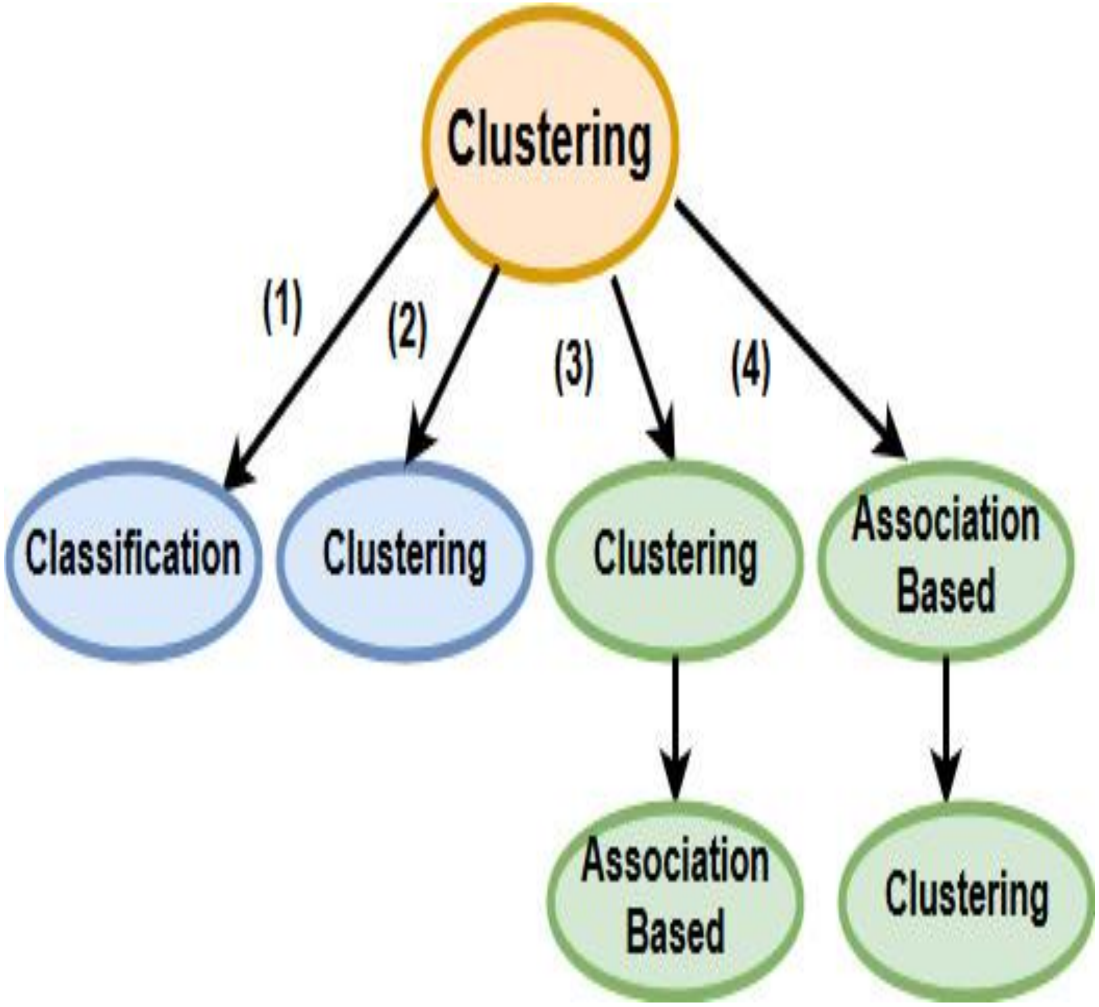


Figure 2.5: Possible Combinations of Clustering Technique with other Data Mining Techniques

As appeared in Figure 2.5 above, four mixes can be structured from clustering. Comparable blends can be produced using association based mining and clustering, henceforth making the absolute conceivable mixes as 12. For instance, in stock administration framework initially out of all things are ordered classification arrangement procedures in predefined classes.

Classes are shaped dependent on the closeness of things dependent on administered learning. In stock administration, classes can be stationary, sustenance, refreshments, and some more. A thing having a place with a particular class is set in that class. A note pad will have a place with stationary class. When the things in stock are arranged, an association is resolved so as to bring the buy conduct of clients. In this manner, association based strategy is utilized to remove these intriguing buy conduct relationship of things.

2.6 Summary

This chapter clarifies the existing state of art considering security aspect in IoT. To do so, first the general mechanism of data collection is discussed in detail using RFID or sensor network as a technology. Later, after data collection the existing security solutions, their differences and their respective weakness and vulnerabilities are identified from the work done in literature. Finally once discussion is made on data collection and its overlay security, the data mining techniques that can be used for extracting the fruitful decisions from the data collected are discussed in IoT.

CHAPTER 3

DESIGNING HYBRID LIGHTWEIGHT LOGICAL SECURITY FRAMEWORK FOR IOT

In this chapter a Hybrid Lightweight Security Framework (HLSF) is designed for IoT. HLSF consists of three phases. First phase is Registration, where every new device that joins the network gets its credentials from the central server. Second phase is Authentication, where every time a device wish to communicate has to prove its identity to server and wants reciprocal identity check for the server. Third phase is Data Security, to offer confidentiality and integrity to the data in transit. Therefore, a Lightweight Data Security (LDS) algorithm is proposed to offer security in HLSF.

3.1 Introduction

A new cryptographic primitive referred as “Lightweight Cryptography” is getting into market specifically for its use in resource constrained environments like IoT using RFID, or sensors as embedded technologies. Lightweight does not intend to be weak in nature, but are not imposed on number of applications. Lightweight algorithms restrict the attacker by exposing only limited data operated by a single key. Lightweight solutions are used for marinating a trade-off between performance, security and resources required.

3.2 Need of Lightweight Cryptography in IoT

IoT consists of resource constrained devices like RFID, sensors that are battery operated. Therefore, a special attention should be given to limit the use of their resources as well as to offer security at same time. Lightweight cryptography solutions offer both security as well as performance. Thinking about resource restriction on IoT, lightweight solutions seems to be best. Lightweight solutions offer security by exposing only limited data in operation. The main reasons behind applying lightweight cryptography in IoT are summarized below:

- **End to End Communication Efficiency:** When two resource constrained devices will communicate using lightweight solutions, the overall energy consumption will reduce. Hence, the end to end communication will be become efficient.

- **Increased Number of Connections:** As lightweight solution requires fewer resources, therefore, any device that is resource constraint can connect to the network. The number of connections in the network will thus increase

3.3 Design Considerations for Lightweight Algorithms

Lightweight block ciphers work well over conventional solutions due to the their lightweight design considerations as specified below:

- **Small Key Size:** Key size opted by lightweight block cipher should be comparatively less than conventional block ciphers. National Institute of Standards and Technology (NIST) imposes a restriction of minimum key size of 112 bits. Key size even less than this more susceptible to brute force attack.

- **Small Block Size:** The block size chosen for the lightweight cipher should be less as compared to the conventional ciphers. For example, if block size is taken as 64 bit as compared to 128 bit of AES, more number of plaintext blocks can be encrypted. Moreover the memory requirements will become less.

- **Simple Round Structure:** The rounds designed for lightweight ciphers should be simple as compared to the conventional cryptographic algorithms. For example, a round can be made simpler by replacing an 8-bit S-Box with a 4- bit S-Box. This will also reduce the memory requirements. This may reduce the level of security that can be improvised by escalating the total number of rounds.

- **Simple Key Schedule:** Generation function of key schedule in lightweight designs must generate the sub keys very fast. Simpler a key schedule is, less power consumption and memory will be required by the algorithm. Using simple key schedule, may lead to attacks like weak key, related key, or chosen key attack but that can be overcome by using a secure and frequent function for key generation.

- **Less Implementation Requirements:** A device should support either encryption or decryption. Only required operations of the cipher should be implemented rather than implementing the full cipher.

By the taking the above parameters into consideration, HLSF is designed where the LSA tends to use the same design considerations as stated above.

3.4 Proposed Hybrid Lightweight Security Framework (HLSF)

The proposed HLSF is divided into three phases- Registration, Authentication, and LDS. When a new device enters into network, it is registered first with the server where it gets the credentials using key sharing mechanism. Once the device has obtained the credentials, before initiating any communication mutual authentication is performed among device and the server. Once device is authenticated, the data communicated to and from the device is made secure using LDS algorithm.

The Notations used in the process of registration, authentication and data security are shown in Table 3.1 below.

Table 3.1: Notations Used in HLSF

Symbol	**Description**
D_i	ith Device
IS	Information Server
ID_D	Identity of device
K_S	Key Shared between Device and Server
SN	Sequence Number
UID	Unique IDs
K_a	Alternate Keys
T_V	Temporary Variable
n	Number of bits in each word
2n	Block size/ Number of Input bits
SK_i	ith Key Sub Block
S^{-x}	Left Rotation by x bits
S^{y}	Right Rotation by y bits

3.4.1 Phase 1: Registration

Steps corresponding to registration phase are detailed below:

Step 1: Device D_i will initiate a connection set up with IS by submitting its ID_D to the IS using a secure medium.

Step 2: IS after getting the connection request from D_i, computes a nonce value N_S. This N_S is used to compute a shared key $K_{S,}$ where $K_S = ID_S \oplus h(ID_D || N_S)$.

Step 3: Along with this, IS generates a set of unique-Ids, UID= $\{id_{1,}\ id_{2,}\ id_3, \ldots . id_n.\}$ and set of alternate keys Ka= $\{k_{a1}, k_{a2}, \ldots . k_{an}\}$relative to each $uid_i \in$UID.

Step 4: Subsequently, IS generates a random sequence number SN. Therefore, for each request sent by the D_i, IS generates K_S, unique Ids, alternate keys, and SN. If D_i makes another request to IS, a new SN is generated. The responsibility of IS is to maintain one copy of SN in database and forward same copy to the D_i. The benefit of using SN is to avoid any kind of replay that can be injected by the intruder.

Step 5: Before the actual authentication starts, IS matches the SN send by D_i with the one stored in the database. If match occurs then authentication Phase-2 becomes active. Whereas, if match does not occur in SN, IS terminates the connection with D_i and asks D_i to use one of the UID and K_a pair.

Once, this pair is used, its entry is removed from the IS database as well as from D_i. At the end IS sends a message to D_i encrypted using public key of D_i having a set of values- K_S, $\{id_i, k_{ai}\}$, SN and also maintains the same values in its own database along with the ID of D_i, that is ID_D.

3.4.2 Phase 2: Authentication

In authentication phase, two way mutual authentication is performed between D_i and IS. Steps corresponding to authentication phase are detailed below:

Step 1: D_i by taking a nonce value N_1, generates a variable $V_1 = h(ID_D || K_S \oplus N_1)$.

Step 2: Now, D_i creates a message of request having $\{V_1, ID_{D,}\ SN\}$ to the IS.

Step 3: In case SN is not available with D_i, D_i will use one of the {id_i, k_{ai}} pair where k_{ai} can be used in replacement of $K_{S.}$

Step 4: On receiving request from D_i, IS verifies the SN in the message, if it matches the SN of corresponding D_i stored in database, it also verifies the other parameters are valid or not. Later, IS computes the value of $N_{1.}$

Step 5: If all the parameters are validated, then IS after taking a nonce value N_2will generate a new random sequence number $SN_{new\ =}$ h (ID_D || K_S || N_1) ⊕ SN and computes a temporary variable T_V = h (ID_D || K_S || N_2) ⊕ SN_{new} and computing variable V_2= h(ID_D || K_S || N_1|| T_V).

Step 6: D_i on receiving message containing {V_2, SN_{New}, T_V} from the IS, computes the value h(ID_D || K_S || N_1|| T_V) and compares it with V_2. If match occurs, D_i computes nonce N_2 using T_V = h (ID_D || K_S || N_2) ⊕ SN_{new}

3.4.3 Phase 3: Lightweight Data Security (LDS) Algorithm

Once mutual authentication is performed between D_i and IS, the next step is to offer data security using encryption method. Data is taken in blocks of 64 bits each, and the size of K_S shared between D_i and IS is 128 bits. To offer security, a Lightweight Data Security (LDS) algorithm is proposed. This algorithm takes of the secure data communication and offers the services for security such as confidentiality of data, integrity of data.

Proposed LDS works on 20 rounds using Addition, Rotation and XOR (ARX) operations. This flexibility of choosing the number of rounds lies with the user depending upon the execution time required and also on full diffusion. The three operations ARX are chosen for offering optimum security trading off with lightweight solution considering the IoT application scenario. The reason for choosing only these operations for a round is discussed later in Section 3.4.3.2. The structure of LDS consisting of 20 Rounds using ARX operations and a key generation function is represented through Figure 3.1.

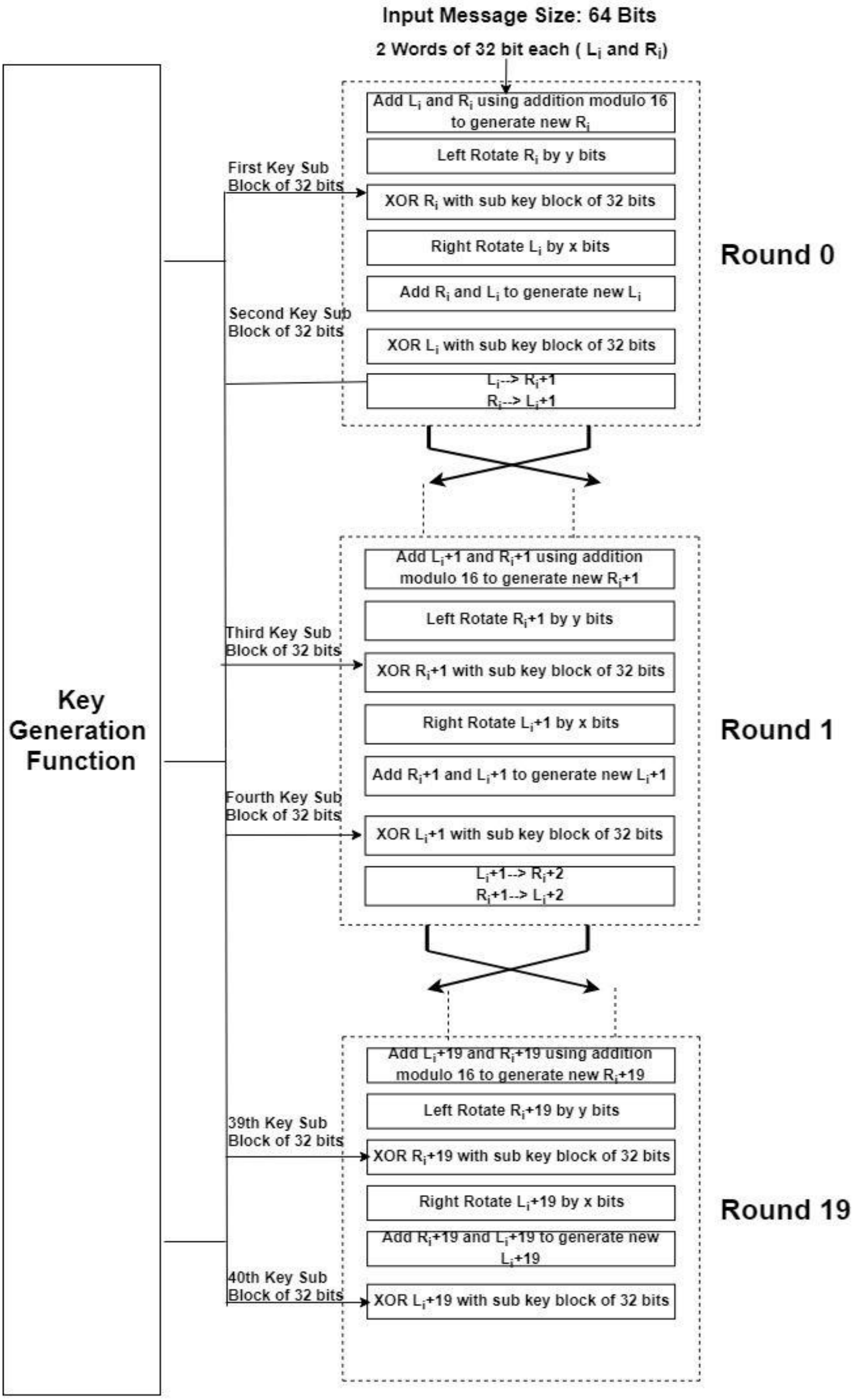

Input Message Size: 64 Bits
2 Words of 32 bit each (L_i and R_i)
Key Generation Function
Add L_i and R_i using addition modulo 16 to generate new R_i
Left Rotate R_i by y bits
First Key Sub Block of 32 bits
XOR R_i with sub key block of 32 bits
Right Rotate L_i by x bits
Round 0
Add R_i and L_i to generate new L_i
Second Key Sub Block of 32 bits
XOR L_i with sub key block of 32 bits
L_i--> R_i+1
R_i--> L_i+1
Add L_i+1 and R_i+1 using addition modulo 16 to generate new R_i+1
Left Rotate R_i+1 by y bits
Third Key Sub Block of 32 bits
XOR R_i+1 with sub key block of 32 bits
Right Rotate L_i+1 by x bits
Round 1
Add R_i+1 and L_i+1 to generate new L_i+1
Fourth Key Sub Block of 32 bits
XOR L_i+1 with sub key block of 32 bits
L_i+1--> R_i+2
R_i+1--> L_i+2
Add L_i+19 and R_i+19 using addition modulo 16 to generate new R_i+19
Left Rotate R_i+19 by y bits
39th Key Sub Block of 32 bits
XOR R_i+19 with sub key block of 32 bits
Right Rotate L_i+19 by x bits
Round 19
Add R_i+19 and L_i+19 to generate new L_i+19
40th Key Sub Block of 32 bits
XOR L_i+19 with sub key block of 32 bits

Figure 3.1: LDS Structure

3.4.3.1 Generation of Sub Keys for Each Round

For each round two n- bit sub key bocks are required, considering n as the number of bits in a word. Block size that can be taken as input will be 2n. Here, block size of 64 bits is assumed, so value of n is 32. Key size is taken as 128 bit. Therefore, for 20 rounds 40 key sub blocks each of 32-bit out of 128 bit long key. A key generation mechanism is required for getting the key sub blocks for each round of operation.

Sub key generation is done in such a manner that key generator gives a unique and random sub key every time it is run. For a good key generator mechanism, if the generated sub key generated is compromised by cryptanalysis, other sub keys should not be identified. The sub keys are generated from the main key of 128 bit. As stated earlier each round requires two sub key blocks. The mechanism of key generation function consists of a key generation that divides the keys into sub blocks. Key generator generates sub keys for two rounds at a time. Therefore, for 20 rounds key generator will work for 10 times and generate 4 sub key blocks each time, making a total of 40 sub key blocks. The whole mechanism of key generator is explained through following steps:

Step 1: The original Key (K_i) of 128 bits is given as input to sub key generator.

Step 2: Sub key generator generates 4 sub key blocks of 32 bit each. Two key sub blocks of 32 bits are passed as input to first round and next two key sub blocks of 32 bits are passed as input to second round.

Step 3: Bits in original K_i are processed using a mixing function to generate input for the running the key generator for the next time. From there again 4 key sub blocks are generated for next two rounds.

Step 4: Mixing Function takes as input the output of the previous key generator function. For the first time, after the execution of key generator, original K_i that consists of 4 key sub blocks, let us say, SK_1, SK_2, SK_3, SK_4, each of 32 bits. Mixing function performs the XOR operation in circular rotation. All the bits of SK_1 are XORed with random bits of SK_2, SK_2 is XORed with random bits of SK_3, SK_3 is XORed with random bits of $SK_{4,}$ SK_4 is XORed with random bits of SK_1. $SK_1(0)$

represents the first bit of key sub block SK_1. Figure 3.2 shows the block diagram for operation of key generator. The sample equations to generate sub key can be represented mathematically as

$$\begin{aligned}
&SK_1(0)= SK_1(0) \oplus SK_2(31) &\qquad& SK_2(0)= SK_2(0) \oplus SK_3(0)\\
&SK_1(1)= SK_1(1) \oplus SK_2(0) &\qquad& SK_2(1)= SK_2(1) \oplus SK_3(1)\\
&SK_1(2)= SK_1(2) \oplus SK_2(1) &\qquad& SK_2(2)= SK_2(2) \oplus SK_3(2)\\
&SK_1(3)= SK_1(3) \oplus SK_2(2) &\qquad& SK_2(3)= SK_2(3) \oplus SK_3(3)\\
&SK_1(4)= SK_1(4) \oplus SK_2(3) &\qquad& SK_2(4)= SK_2(4) \oplus SK_3(4)\\
&SK_1(5)= SK_1(5) \oplus SK_2(4) &\qquad& SK_2(5)= SK_2(5) \oplus SK_3(5)\\
&SK_1(6)= SK_1(6) \oplus SK_2(5) &\qquad& SK_2(6)= SK_2(6) \oplus SK_3(6)\\
&SK_1(7)= SK_1(7) \oplus SK_2(6) &\qquad& SK_2(7)= SK_2(7) \oplus SK_3(7)\\
&SK_1(8)= SK_1(8) \oplus SK_2(7) &\qquad& SK_2(8)= SK_2(8) \oplus SK_3(8)\\
&SK_1(9)= SK_1(9) \oplus SK_2(8) &\qquad& SK_2(9)= SK_2(9) \oplus SK_3(9)\\
&SK_1(10)= SK_1(10) \oplus SK_2(9) &\qquad& SK_2(10)= SK_2(10) \oplus SK_3(10)\\
&SK_1(11)= SK_1(11) \oplus SK_2(10) &\qquad& SK_2(11)= SK_2(11) \oplus SK_3(11)\\
&SK_1(12)= SK_1(12) \oplus SK_2(11) &\qquad& SK_2(12)= SK_2(12) \oplus SK_3(12)\\
&SK_1(13)= SK_1(13) \oplus SK_2(12) &\qquad& SK_2(13)= SK_2(13) \oplus SK_3(13)\\
&SK_1(14)= SK_1(14) \oplus SK_2(13) &\qquad& SK_2(14)= SK_2(14) \oplus SK_3(14)\\
&SK_1(15)= SK_1(15) \oplus SK_2(14) &\qquad& SK_2(15)= SK_2(15) \oplus SK_3(15)\\
&SK_1(16)= SK_1(16) \oplus SK_2(15) &\qquad& SK_2(16)= SK_2(16) \oplus SK_3(16)\\
&SK_1(17)= SK_1(17) \oplus SK_2(16) &\qquad& SK_2(17)= SK_2(17) \oplus SK_3(17)\\
&SK_1(18)= SK_1(18) \oplus SK_2(17) &\qquad& SK_2(18)= SK_2(18) \oplus SK_3(18)\\
&SK_1(19)= SK_1(19) \oplus SK_2(18) &\qquad& SK_2(19)= SK_2(19) \oplus SK_3(19)\\
&SK_1(20)= SK_1(20) \oplus SK_2(19) &\qquad& SK_2(20)= SK_2(20) \oplus SK_3(20)\\
&SK_1(21)= SK_1(21) \oplus SK_2(20) &\qquad& SK_2(21)= SK_2(21) \oplus SK_3(21)\\
&SK_1(22)= SK_1(22) \oplus SK_2(21) &\qquad& SK_2(22)= SK_2(22) \oplus SK_3(22)\\
&SK_1(23)= SK_1(23) \oplus SK_2(22) &\qquad& SK_2(23)= SK_2(23) \oplus SK_3(23)\\
&SK_1(24)= SK_1(24) \oplus SK_2(23) &\qquad& SK_2(24)= SK_2(24) \oplus SK_3(24)\\
&SK_1(25)= SK_1(25) \oplus SK_2(24) &\qquad& SK_2(25)= SK_2(25) \oplus SK_3(25)\\
&SK_1(26)= SK_1(26) \oplus SK_2(25) &\qquad& SK_2(26)= SK_2(26) \oplus SK_3(26)\\
&SK_1(27)= SK_1(27) \oplus SK_2(26) &\qquad& SK_2(27)= SK_2(27) \oplus SK_3(27)\\
&SK_1(28)= SK_1(28) \oplus SK_2(27) &\qquad& SK_2(28)= SK_2(28) \oplus SK_3(28)
\end{aligned}$$

$SK_1(29)= SK_1(29) \oplus SK_2(28)$	$SK_2(29)= SK_2(29) \oplus SK_3(29)$
$SK_1(30)= SK_1(30) \oplus SK_2(29)$	$SK_2(30)= SK_2(30) \oplus SK_3(30)$
$SK_1(31)= SK_1(31) \oplus SK_2(30)$	$SK_2(31)= SK_2(31) \oplus SK_3(31)$
$SK_3(0)= SK_3(0) \oplus SK_4(1)$	$SK_4(0)= SK_4(0) \oplus SK_1(2)$
$SK_3(1)= SK_3(1) \oplus SK_4(2)$	$SK_4(1)= SK_4(1) \oplus SK_1(3)$
$SK_3(2)= SK_3(2) \oplus SK_4(3)$	$SK_4(2)= SK_4(2) \oplus SK_1(4)$
$SK_3(3)= SK_3(3) \oplus SK_4(4)$	$SK_4(3)= SK_4(3) \oplus SK_1(5)$
$SK_3(4)= SK_3(4) \oplus SK_4(5)$	$SK_4(4)= SK_4(4) \oplus SK_1(6)$
$SK_3(5)= SK_3(5) \oplus SK_4(6)$	$SK_4(5)= SK_4(5) \oplus SK_1(7)$
$SK_3(6)= SK_3(6) \oplus SK_4(7)$	$SK_4(6)= SK_4(6) \oplus SK_1(8)$
$SK_3(7)= SK_3(7) \oplus SK_4(8)$	$SK_4(7)= SK_4(7) \oplus SK_1(9)$
$SK_3(8)= SK_3(8) \oplus SK_4(9)$	$SK_4(8)= SK_4(8) \oplus SK_1(10)$
$SK_3(9)= SK_3(9) \oplus SK_4(10)$	$SK_4(9)= SK_4(9) \oplus SK_1(11)$
$SK_3(10)= SK_3(10) \oplus SK_4(11)$	$SK_4(10)= SK_4(10) \oplus SK_1(12)$
$SK_3(11)= SK_3(11) \oplus SK_4(12)$	$SK_4(11)= SK_4(11) \oplus SK_1(13)$
$SK_3(12)= SK_3(12) \oplus SK_4(13)$	$SK_4(12)= SK_4(12) \oplus SK_1(14)$
$SK_3(13)= SK_3(13) \oplus SK_4(14)$	$SK_4(13)= SK_4(13) \oplus SK_1(15)$
$SK_3(14)= SK_3(14) \oplus SK_4(15)$	$SK_4(14)= SK_4(14) \oplus SK_1(16)$
$SK_3(15)= SK_3(15) \oplus SK_4(16)$	$SK_4(15)= SK_4(15) \oplus SK_1(17)$
$SK_3(16)= SK_3(16) \oplus SK_4(17)$	$SK_4(16)= SK_4(16) \oplus SK_1(18)$
$SK_3(17)= SK_3(17) \oplus SK_4(18)$	$SK_4(17)= SK_4(17) \oplus SK_1(19)$
$SK_3(18)= SK_3(18) \oplus SK_4(19)$	$SK_4(18)= SK_4(18) \oplus SK_1(20)$
$SK_3(19)= SK_3(19) \oplus SK_4(20)$	$SK_4(19)= SK_4(19) \oplus SK_1(21)$
$SK_3(20)= SK_3(20) \oplus SK_4(21)$	$SK_4(20)= SK_4(20) \oplus SK_1(22)$
$SK_3(21)= SK_3(21) \oplus SK_4(22)$	$SK_4(21)= SK_4(21) \oplus SK_1(23)$
$SK_3(22)= SK_3(22) \oplus SK_4(23)$	$SK_4(22)= SK_4(22) \oplus SK_1(24)$
$SK_3(23)= SK_3(23) \oplus SK_4(24)$	$SK_4(23)= SK_4(23) \oplus SK_1(25)$
$SK_3(24)= SK_3(24) \oplus SK_4(25)$	$SK_4(24)= SK_4(24) \oplus SK_1(26)$
$SK_3(25)= SK_3(25) \oplus SK_4(26)$	$SK_4(25)= SK_4(25) \oplus SK_1(27)$
$SK_3(26)= SK_3(26) \oplus SK_4(27)$	$SK_4(26)= SK_4(26) \oplus SK_1(28)$
$SK_3(27)= SK_3(27) \oplus SK_4(28)$	$SK_4(27)= SK_4(27) \oplus SK_1(29)$
$SK_3(28)= SK_3(28) \oplus SK_4(29)$	$SK_4(28)= SK_4(28) \oplus SK_1(30)$
$SK_3(29)= SK_3(29) \oplus SK_4(30)$	$SK_4(29)= SK_4(29) \oplus SK_1(31)$
$SK_3(30)= SK_3(30) \oplus SK_4(31)$	$SK_4(30)= SK_4(30) \oplus SK_1(0)$
$SK_3(31)= SK_3(31) \oplus SK_4(0)$	$SK_4(31)= SK_4(31) \oplus SK_1(1)$

Step 5: Step 3 and Step 4 are repeated till all the 40 sub key blocks are generated for all the 20 rounds.

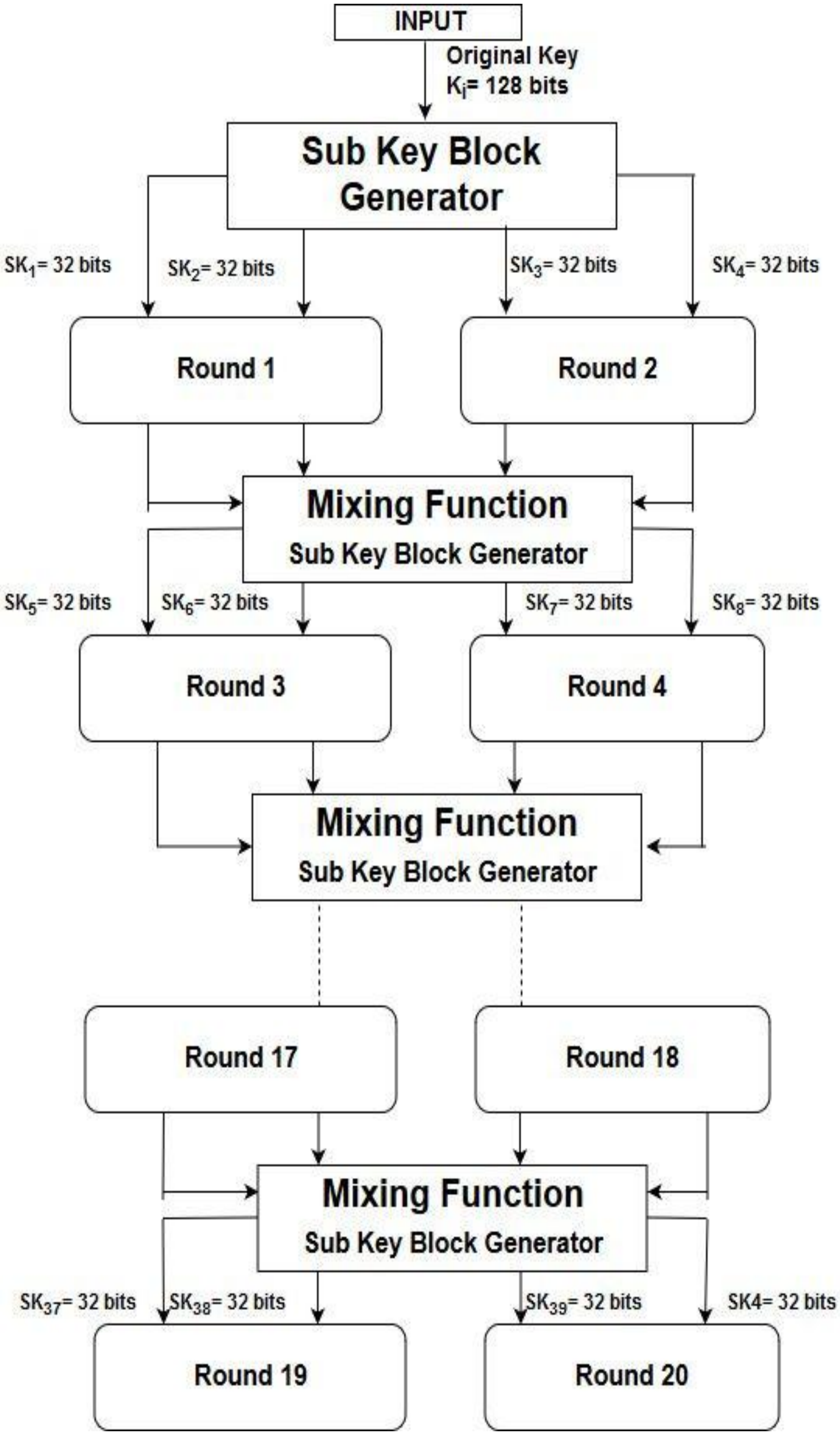


Figure 3.2: Block Diagram for Key Sub Block Generator Function

3.4.3.2 Round Function of LDS

LDS framework works on Feistel like structure. Operations used during the encryption process of LDS are

- **Addition modulo 2^n**, considering n as the number of bits in a word. If n is 16, block size will be 32 and for n taken as 32, block size will be 64 bits. Addition modulo is preferred over multiplication modulo. There may be multiple reasons for choosing addition over multiplication. First, multiplication require more cycles as compared to addition even with the fastest CPUs. Second, operation of multiplication may lead to timing attacks.

- **Bitwise XOR, ⊕:** Most block ciphers work using XOR as the basic operation as compared to other operations like AND and OR. Numbers of factors supporting XOR over other operations are; First, XOR operation works on reversible procedure. When encryption is performed on original text XOR with key to generate cipher text, same key when operated using XOR with cipher text the resultant will be same original text. Second, XOR can be realized using the NAND gate requiring few transistors as compared to other operations, making its hardware implementation quite easier. Third, in XOR the output is dependent on both the operands as compared to AND and OR. In AND, if one of the operand is false, second is not evaluated at all. In OR, if one of the operand is true second is not evaluated at all. Whereas, in XOR if first operand is true or false, second need to be evaluated for getting the expected output.

- **Rotations Left and Right**, R^{-b} and R^{b} respectively, where b is the number of bits to rotate. Rotations are preferred over shift as rotation when used with the XOR operation creates maximum diffusion in the resultant output with alteration in a single input bit. On the other hand, when shift is used with the XOR then diffusion created is less in output with alteration in a single input bit.

The input block of n bits is divided into two equal halves. For example, if input text is 64 bit long, it will be divided into 32 bit each represented as L_i and R_i. L_i represents the left sub block and R_i represents the right sub block. The left and the right sub block in a particular round is evaluated as

$$L_i = (S^{-x} L_i + (S^y (L_i + R_i) \oplus SK_2)) \oplus SK_1$$

$$R_i = S^y (L_i + R_i) \oplus SK_2$$

Therefore, the round function of LDS is denoted as

$$\mathbf{F(L_i, R_i) = ((S^{-x} L_i + (S^y (L_i + R_i) \oplus SK_2)) \oplus SK_1, \; S^y (L_i + R_i) \oplus SK_2)}$$

Where x and y are the rotation constants. For block size of 64 bit and key size of 128 bits, the value of x is taken as 7 and the value of y is taken as 3. This composition of round function is represented through Figure 3.3 below

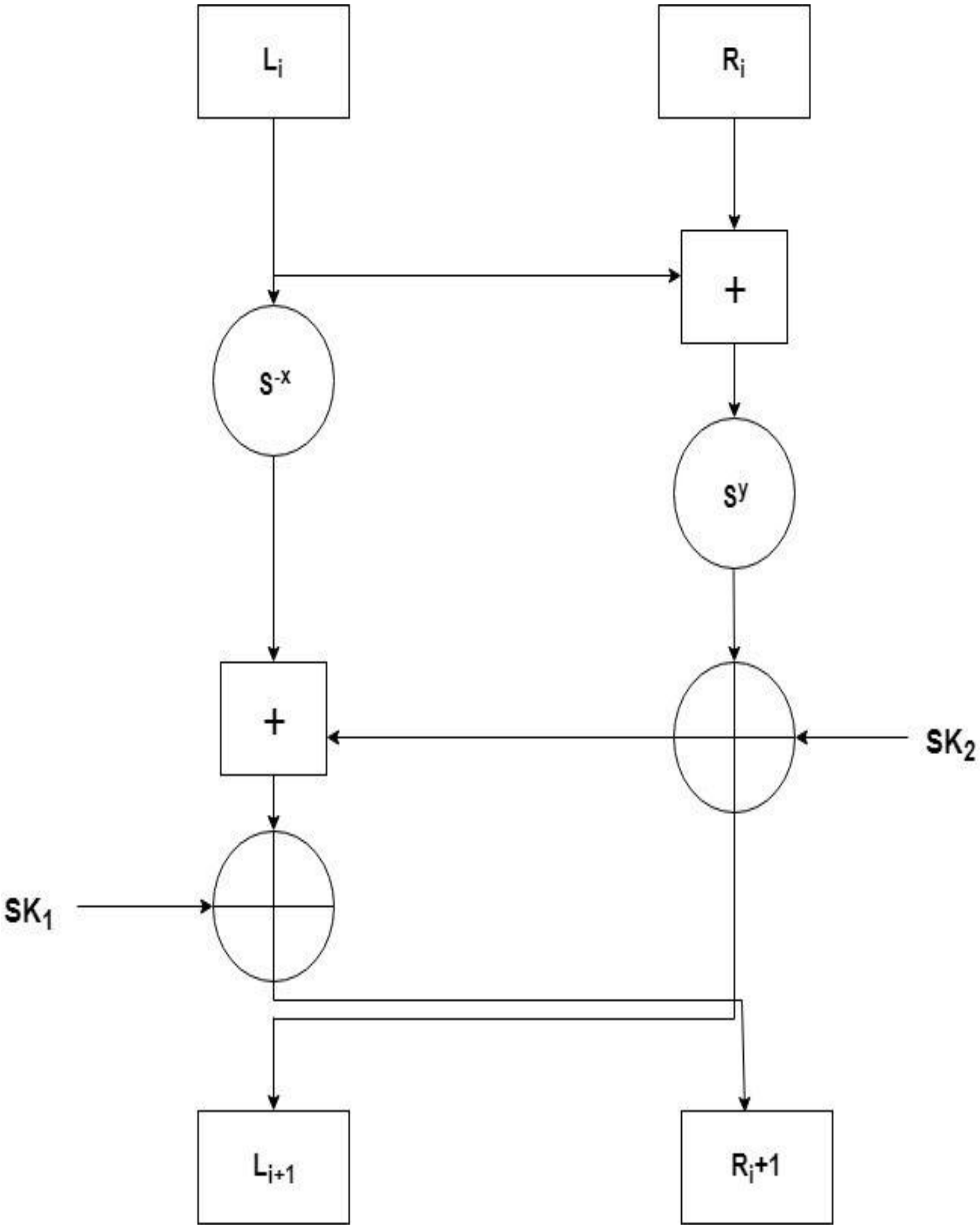


Figure 3.3: Single Round Function of LDS

3.4.3.3 Evaluating Diffusion Property for LDS Round

Diffusion property of cryptographic algorithm is focused on incorporating the avalanche effect. It refers to observe the change in number of bits of output cipher text with a single bit modification in the input original text. With more number of bits affected by diffusion, the cryptographic solution proves to be stronger.

The input original text of 64 bits is divided into 2 sub data blocks of 32 bits each and referred as A and B. The values of rotation constant for creating the diffusion matrix may vary from 0 to 31. Therefore, the total possible combinations of rotation constants x and y each carrying value from 0 to 31 may lead to 32*32=1024 combinations. 1024 combinations can generate 1024 diffusion matrix based on respective designs. A sample diffusion matrix is shown in Table 3.2 below:

Table 3.2: Generalized Diffusion Table

Input \ Output	Left Block (A)	Right Block (B)
Left Block (A)	M_{AA}	M_{AB}
Right Block (B)	M_{BA}	M_{BB}

In the above table M_{AA} refers to the count of number of bits modified in A by modifying the single bit of A. As there are 32 bits in A, the mean value and the standard deviation is calculated after changing every bit of A and noticing its effect on A and B. Similar effect can be noticed in M_{AB}, M_{BA}, and M_{BB}.

In order to generate the diffusion matrix, certain steps are taken that require the input, rotation constants assumed as x and y here.

Step 1: Input block of 64 bits is divided into two sub blocks referred as A and B each of 32 bits.

Step 2: Considering the different combinations of x and y where each can take values from 0 to 31. Thus, the overall possible combinations are 1024. Here the value of x and y is assumed to be fixed that is x is taken as 7 and y is taken as 3, while executing the LDS algorithm.

Step 3: Round function of LDS is executed over the data blocks A and B to generate output as A^1 and B^1.

Step 4: Start by modifying one bit of A, then execute LDS over modified A bits and the original B to get output as A^2 and B^2. Compare bits of A^1 with bits of A^2 and calculate the number of bits which have been altered, that will become value of M_{AA}. Similarly compare bits of B^1 with bits of B^2 and calculate the number of bits which have been altered, that will become value of M_{AB}.

Step 5: Repeat Step 3 but this time by modifying one bit of B. Execute LDS over modified B bits and the original A to get output as A^2 and B^2. Compare bits of A^1 with bits of A^2 and calculate the number of bits which have been altered, that will become the value of M_{BA}. Similarly compare bits of B^1 with bits of B^2 and calculate the number of bits which have been altered, that will become the value of M_{BB}.

Step 6: Repeat steps 3 and 4 at least 64 times using rotation constant x=7 and y=3 to find the average of each value of matrix M to get diffusion table as shown in Table 3.3 below.

Table 3.3: Diffusion Table Considering x=7 and y=3

Input \ **Output**	**Left Block (A)**	**Right Block (B)**
Left Block (A)	10.5	12.3125
Right Block (B)	6.71815	8.90625
Mean= 9.609, Standard Deviation= 2.058		

The possible combinations for and y can be 1024, but the same steps for creating diffusion table are repeated by considering the most common used rotation constants x=8 and y=3 and second time by considering x=7 and y=2. The diffusion table generated by repeating the steps 64 times for the rotation constants x=8 and y=3 is shown through Table 3.4 below

Table 3.4: Diffusion Table Considering x=8 and y=3

Input \ Output	**Left Block (A)**	**Right Block (B)**
Left Block (A)	9.375	10.156
Right Block (B)	3.156	7.218
Mean= 7.4762, Standard Deviation= 2.716		

The diffusion table generated by repeating the steps 64 times for the rotation constants x=7 and y=2 ix shown through Table 3.5 below

Table 3.5: Diffusion Table Considering x=7 and y=2

Input \ Output	**Left Block (A)**	**Right Block (B)**
Left Block (A)	8.625	11.25
Right Block (B)	4.312	6.75
Mean= 7.734, Standard Deviation= 2.541		

Mean value in all the combinations shows the average number of bits that are affected by changing the individual bits as shown in equation 8

$$\text{Mean} = (\text{MAA} + \text{MAB} + \text{MBB} + \text{MBA})/\,4 \quad (8)$$

Standard Deviation is calculated by finding the variance after subtracting each data value from the mean and then finding their sum and finally performing square root as shown in equation 9.

$$\text{Standard Deviation} = \sqrt{\sum_{i,j=1}^{2}}(\text{Mij} - Mean) \quad (9)$$

The mean and standard deviation for three sets of rotation constants are shown in Table 3.6 and is represented through Figure 3.4

Table 3.6: Mean and Standard Deviation with different sets of Rotation Constants

Rotation Constants	Mean	Standard Deviation
x=7 **y=3**	**9.609**	**2.058**

x=8
y=3
	7.476	2.716

x=7
y=2
	7.734	2.541

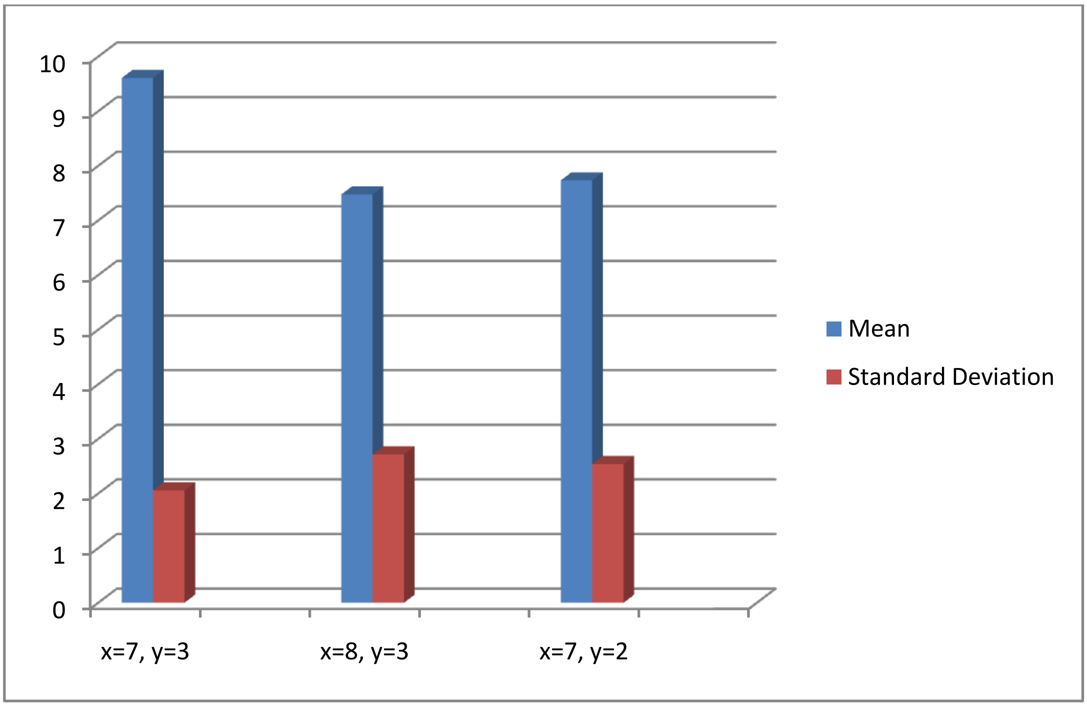


Figure 3.4: Mean and Standard Deviation for different sets of Rotation Constants

Figure 3.4 above clarifies that when same LDS is executed with different combinations of rotation constants, the maximum value and the least standard deviation is observed with rotation constants x=7 and y=3 . Therefore, the proposed LDS algorithm chooses the rotation constants x=7 and y=3.

3.5 Summary

Once data is collected through devices using sensors, the next concern is to offer security to data or devices that play active role in communication. Therefore, this chapter proposes a complete HLSF consisting of three phases. Phase 1 registers the new devices with the central server and handover the essential credentials to the device. Phase 2, performs mutual authentication between server and the device. Phase 3, proposed a LDS algorithm that offers confidentiality and integrity to data in transit. LDS works as a feistel structure on 20 rounds of operation using ARX operations-addition, rotation and XOR.

CHAPTER 4

PERFORMANCE COMPARISON OF EXISTING LIGHTWEIGHT SECURITY ALGORITHMS WITH PROPOSED DATA SECURITY ALGORITHM

This chapter initially highlights the features of the tool used for implementation, that is, COOJA. Later, comparison is made among the existing lightweight security algorithms such as SIMON, SPECK, FANTOMAS and TWINE in COOJA with the proposed LDS on the basis of execution time, memory requirements, throughput, avalanche effect, and energy efficiency.

4.1 Introduction

Now-a-days IoT is admitting in homes, work places, social places or in business firms that will open doors for security and privacy challenges. So, security and privacy issues are becoming major reasons of concern in operation of IoT. The amount of loss that can occur is prominent to imagine if any attack is injected in IoT. Various attacks on IoT exist like eavesdropping, spoofing, DoS, replay attacks, false signals injection. These attacks will tear down the security services of IoT like confidentiality, integrity, and authentication; moreover, it will impact the privacy of users. IoT provides inbuilt primitive security solutions at each layer, which are still vulnerable to attacks.

Traditional cryptography and authentication schemes do not fit well in IoT scenario due to its constrained resources like power, real time execution. So, lightweight cryptography algorithms tend to work well in IoT. Number of lightweight Symmetric and Asymmetric cryptography algorithms exists in literature like AES, HIGHT, RC5, PRESENT, RSA, ECC and many more. These existing algorithms do not guarantee an optimum level of security in real time communication due to more execution time, code length, and memory requirements. Execution time includes time for key management and distribution, encryption and decryption that decides the effectiveness of the protocol.

Asymmetric algorithms are slow due to their large key size, whereas symmetric algorithms can provide only confidentiality and integrity but no authentication leading to attack on availability. This can affect real time information collecting and processing and will fritter away the resources of IoT. This calls for a secure algorithm for IoT that will guarantee services like confidentially, integrity and authentication in optimal time.

Next section highlights the effectiveness of tool used for evaluating the performance of existing traditional as well as modernized lightweight existing algorithms and the comparing it with proposed LDS for IoT by finding how energy efficient they are, their memory requirements, and their execution times. This evaluation is done in COOJA simulator working on Contiki OS.

4.2 Tool Used and Its Features

Contiki comes with a sophisticated Wireless Sensor Network Simulator, COOJA, which is based on real hardware emulation and time-accurate WSN simulation. It can be compiled on linux pc or on virtual environment like VMWare on windows for establishing sensor motes. It is an extensible java based simulator for emulating different nodes like Tmote, sky etc. The code is mostly written in standard C language then upload to target mode. Thus it makes fast and easy for rapid development. Contiki OS supports fully standardized IPv6 and IPv4 internet standards like 6Lowpan, RPL, CoAP etc. Foremost it as a open source software which is available online easily. Following are the features included in Contiki are mentioned below.

- CoAP is a lightweight HTTP protocol that so interaction with the IoT sensors nodes for read and write purpose. COOJA helps to establish a network which is compatible with its protocol.
- COOJA establish RESTful Protocol which uses actions like GET, POST, PUT, DELETE, OBSERVE, DISCOVER in CoAP.
- COOJA helps to establish particular motes (sensors) with IPv6 network.
- COOJA execute CoAP output on browser from where the parameters can be controlled. Example coap://[aaaa::212:7402:2:202] or any other sensor mote

- Program is written in C language so it is easy to deploy in targeted motes.
- The codes that were executed by the nodes is the exact same firmware which can be later on may uploaded to the physical devices.
- It can simulate motes from smaller to larger network.
- COOJA gives a GUI on which we can observe how various motes interact with each other in the targeted environment.
- On output window we can observe network simulation, simulation control, notes, motes output, timeline of motes etc.

4.2.1 Simulation Setup: Contiki OS and the COOJA Simulator

In this section Contiki OS and the COOJA simulation [127] along with the over view of the protocol stack used in the simulation is introduced. This section also cover configuration of the nodes on which the different scenarios network topologies were tested.

The open source operation system, Contiki implements over the IPv6 network involving constrained devices. It shows in Contiki OS application layer for CoAP implementation, Erbium CoAP is used and for the routing of IPv6 network it includes Contiki routing protocol (RPL)[128][129]. In Contiki 2.7, network formation relies on RPL working as integral part of IPv6 Stack. RPL is used for network topology maintenance as well as for packet routing. COOJA offered by Contiki works on graphical interfaces for interaction and thus making it user friendly.

COOJA has inbuilt capabilities of adding the dynamic motes, to fix the transmission range of motes. COOJA implicitly provides mechanisms for data collection by motes and implicit mechanism of security using CoAP at the application layer. The user interface offered by COOJA displays clearly the content, for adding motes, creating scenario, the different tools available for modifying the simulation set up settings. Figure 4.1 shows the differences between the IEFT protocol stack and Contiki protocol stack.

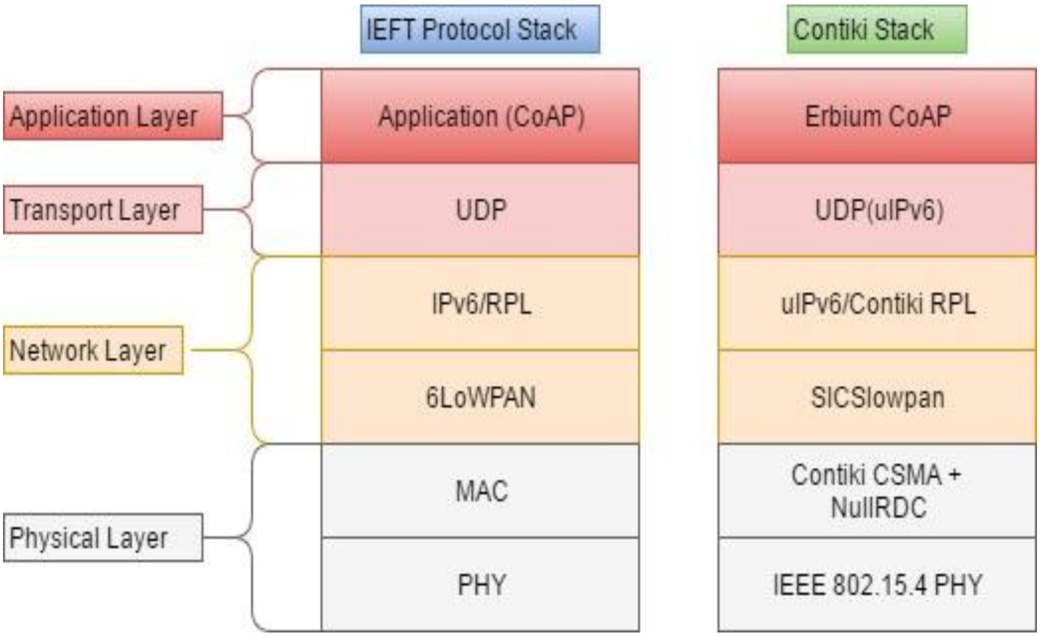
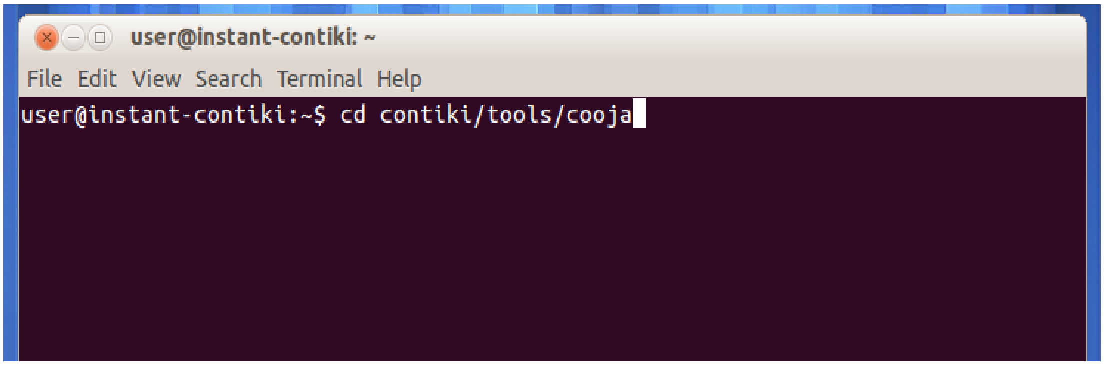


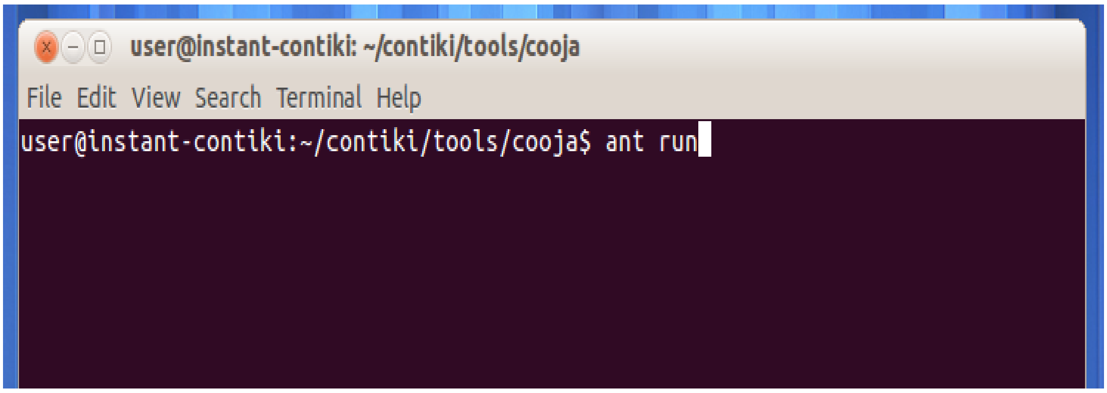
Figure 4.1 Difference Between IEFT Protocol Stack and Contiki OS Stack

Figure 4.2 and Figure 4.3 represents the commands required to initialize COOJA and Figure 4.4 shows the Graphical User Interface (GUI) of COOJA at start up.



```
user@instant-contiki:~$ cd contiki/tools/cooja
```

Figure 4.2: Running COOJA


user@instant-contiki: ~/contiki/tools/cooja
File Edit View Search Terminal Help

```
user@instant-contiki:~/contiki/tools/cooja$ ant run
```

Figure 4.3: Start COOJA with Given Command

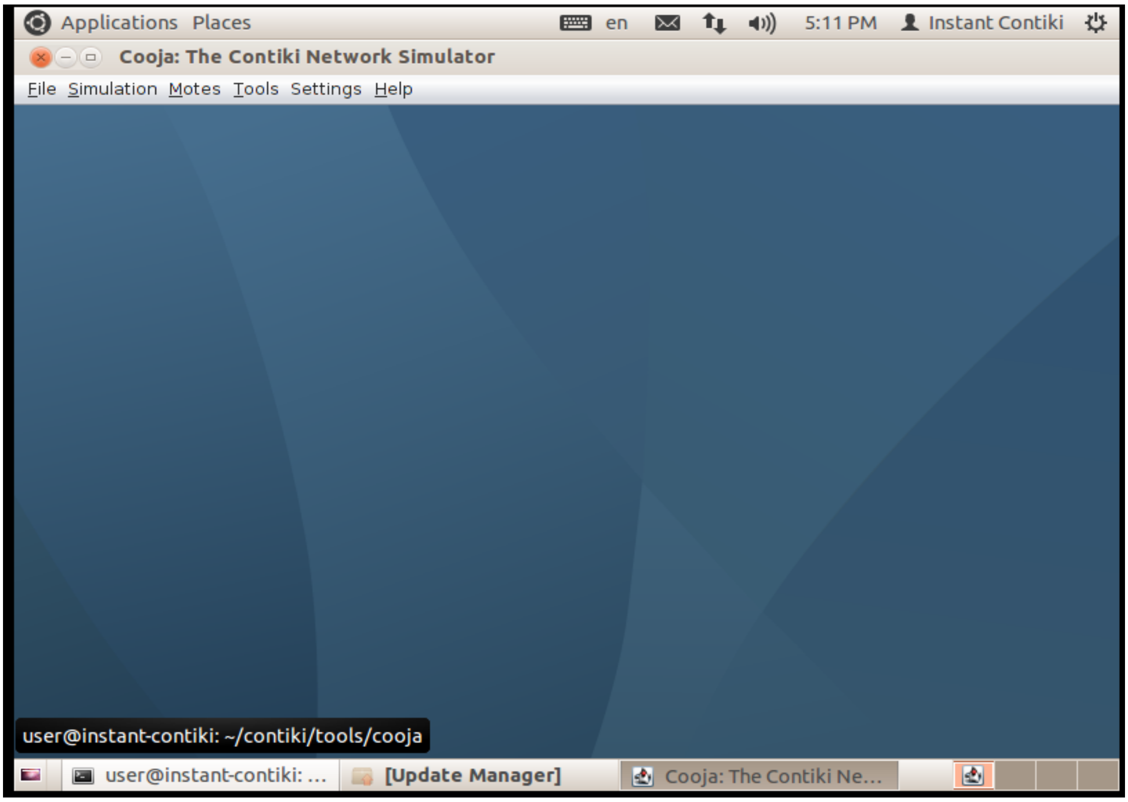

Applications Places
en
5:11 PM
Instant Contiki
Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
user@instant-contiki: ~/contiki/tools/cooja
user@instant-contiki: ...
[Update Manager]
Cooja: The Contiki Ne...

Figure 4.4: COOJA GUI Interface

4.2.2 Simulation Parameters

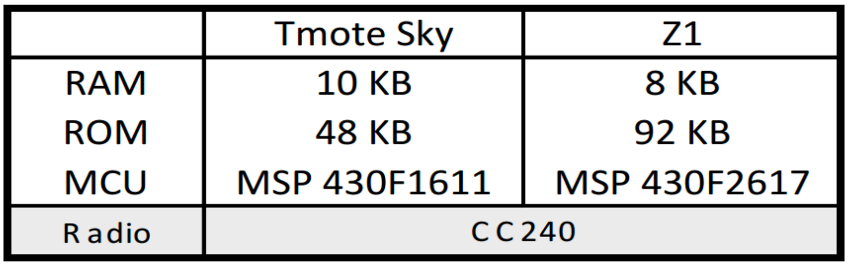
Following are the simulation parameters opted for creating the simulation set up and for network operations.

Parameter	**Range**
Processor	Intel Celeron
RAM Configuration	2 GB
Mote Interfaces	15
Radio medium	UDGM
Transmission range	50 m
Mode Start up Delay	1000 ms
Nodes Position	Random
Sensing and reporting interval	10 s
MAC protocol	Contiki MAC
Routing protocol	RPL

4.2.3 Sensor Mote and Network Configuration

COOJA provide us the software which is enough to emulate the real sensor node over the real hardware. Thus it provides the real memory and processing capabilities in the simulation environment which is in real hardware nodes. For simulating CoAP over application layer COOJA provides IEEE 802.15.4 capable node for radio transceiver (CC 2420) for simulation are: Z1 from Zolertia and Tmote Sky from Mote iv which is similar to TesloB mote for real hardware scenario. Following Table 4.1 describe the RAM, ROM and MCU capabilities of the mode used in the simulation.

Table 4.1: Hardware specification of Tmote Sky and Z1 mote



	Tmote Sky	Z1
RAM	10 KB	8 KB
ROM	48 KB	92 KB
MCU	MSP 430F1611	MSP 430F2617
Radio	CC 240	

Figure 4.5 shows the way to add new motes in the simulation set up. Nodes are added and are placed at random position. 30 nodes are taken in the experimental evaluation as represented in Figure 4.6 below. Figure 4.7 to Figure 4.17 represents motes sending and receiving, communication diagram, Radio Message Analyze, Timelines, network with nodes position in different variations.

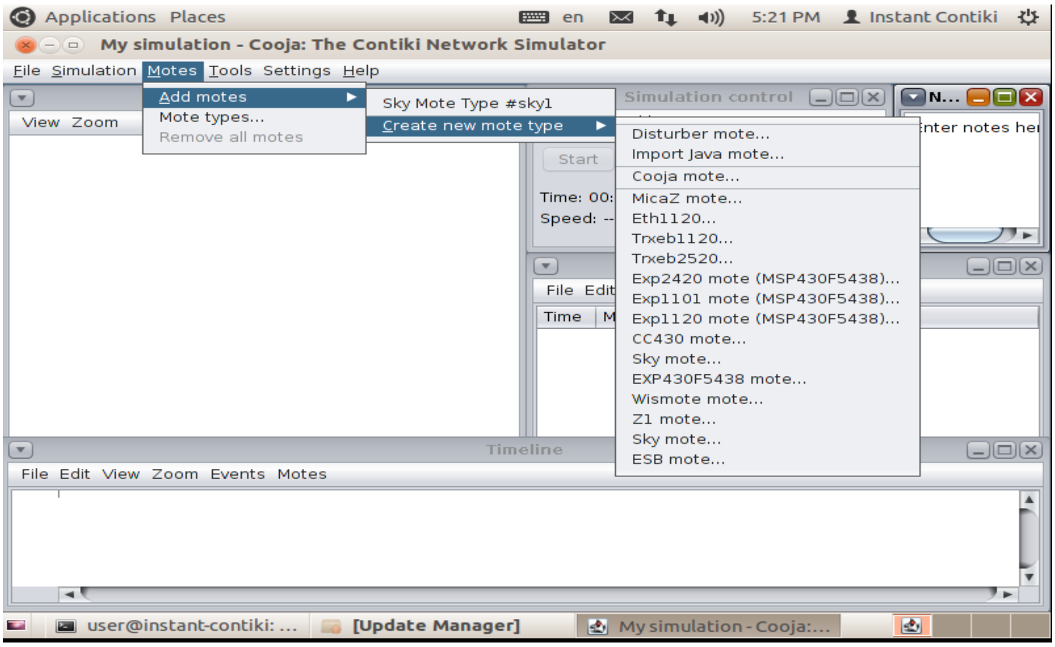


Figure 4.5: Adding Motes

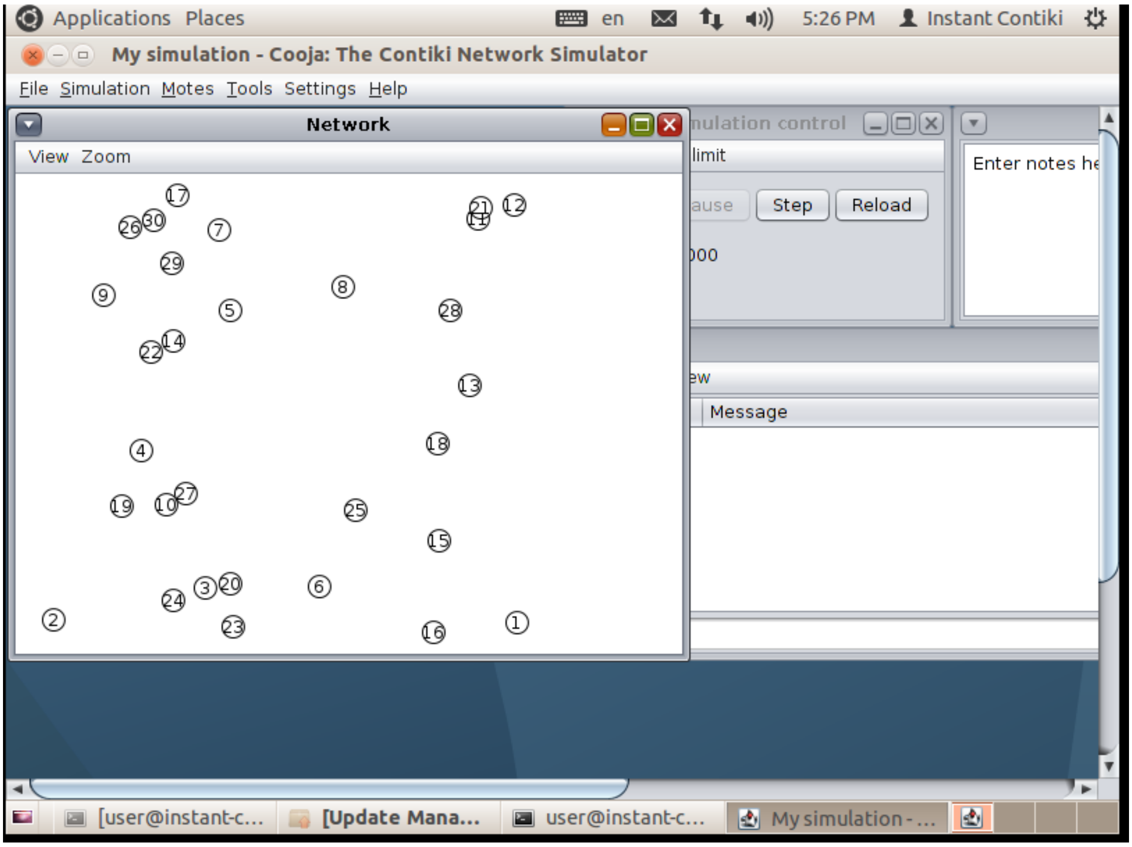
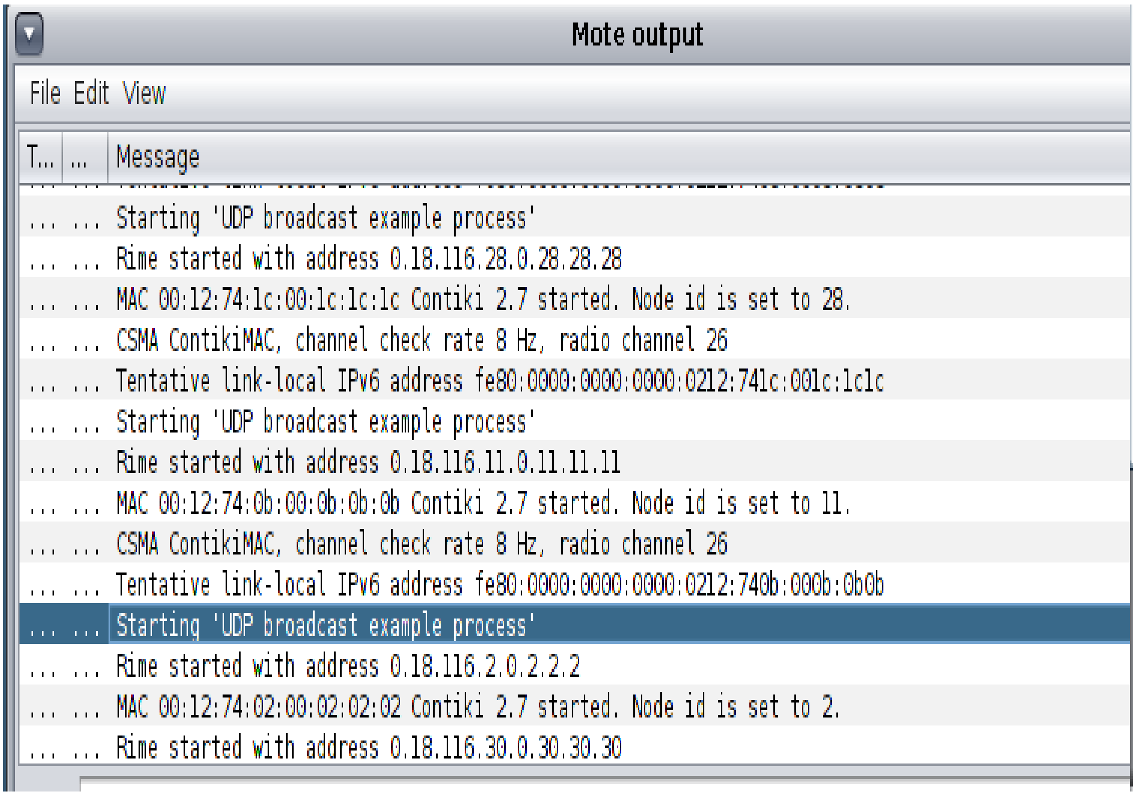
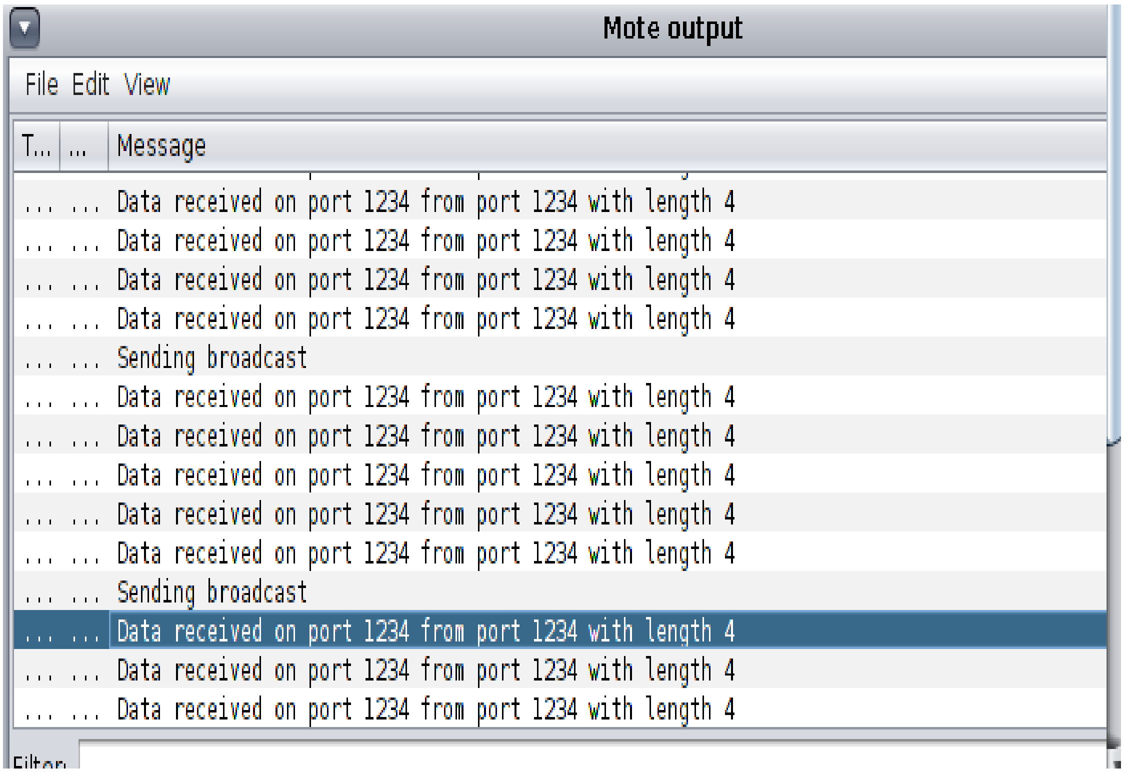


Figure 4.6: Adding 30 motes at Random Position



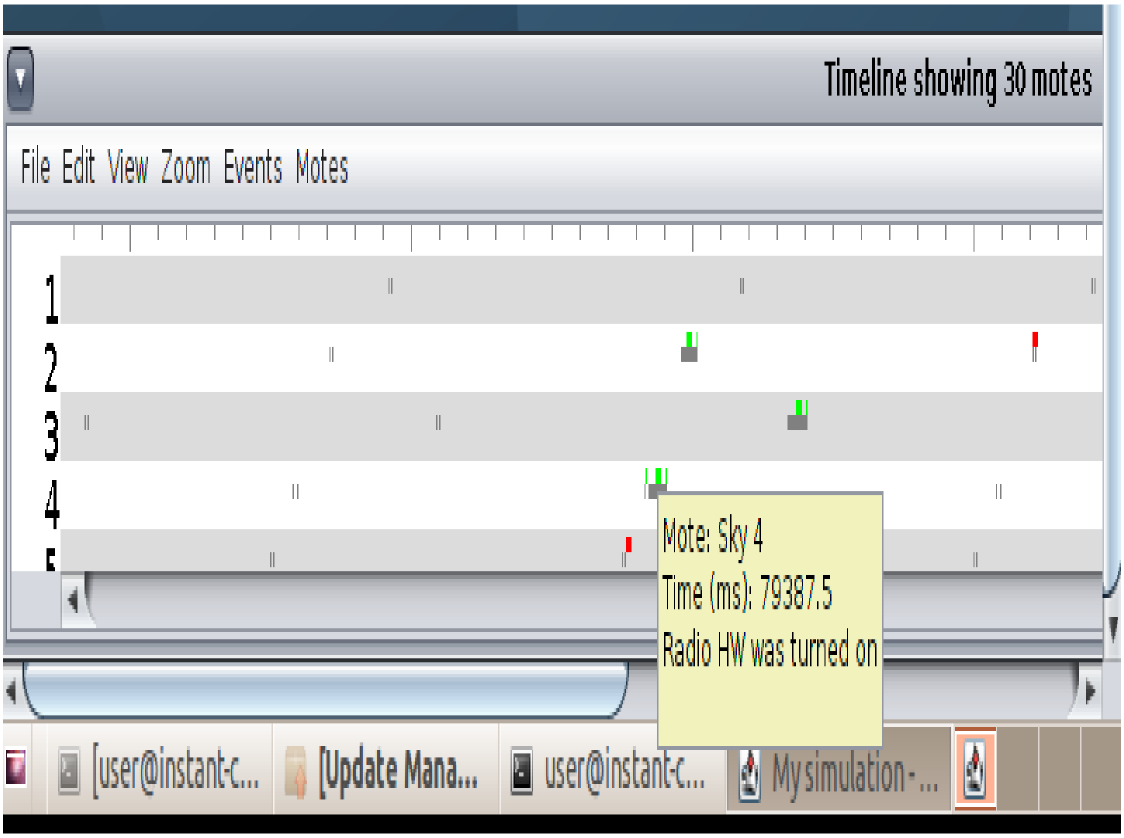
```
... ... Starting 'UDP broadcast example process'
... ... Rime started with address 0.18.116.28.0.28.28.28
... ... MAC 00:12:74:1c:00:1c:1c:1c Contiki 2.7 started. Node id is set to 28.
... ... CSMA ContikiMAC, channel check rate 8 Hz, radio channel 26
... ... Tentative link-local IPv6 address fe80:0000:0000:0000:0212:741c:001c:1c1c
... ... Starting 'UDP broadcast example process'
... ... Rime started with address 0.18.116.11.0.11.11.11
... ... MAC 00:12:74:0b:00:0b:0b:0b Contiki 2.7 started. Node id is set to 11.
... ... CSMA ContikiMAC, channel check rate 8 Hz, radio channel 26
... ... Tentative link-local IPv6 address fe80:0000:0000:0000:0212:740b:000b:0b0b
... ... Starting 'UDP broadcast example process'
... ... Rime started with address 0.18.116.2.0.2.2.2
... ... MAC 00:12:74:02:00:02:02:02 Contiki 2.7 started. Node id is set to 2.
... ... Rime started with address 0.18.116.30.0.30.30.30
```

Figure 4.7: Nodes Initial Output



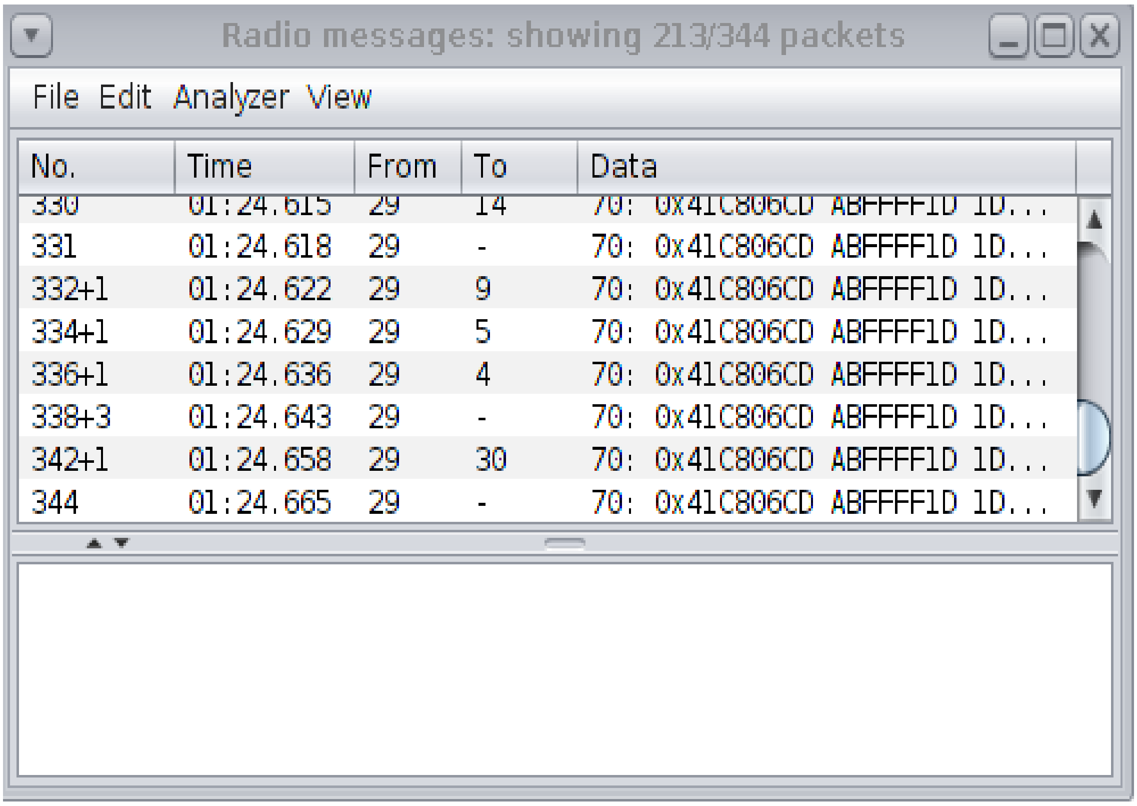
```
T... ...  Message
... ...  Data received on port 1234 from port 1234 with length 4
... ...  Data received on port 1234 from port 1234 with length 4
... ...  Data received on port 1234 from port 1234 with length 4
... ...  Data received on port 1234 from port 1234 with length 4
... ...  Sending broadcast
... ...  Data received on port 1234 from port 1234 with length 4
... ...  Data received on port 1234 from port 1234 with length 4
... ...  Data received on port 1234 from port 1234 with length 4
... ...  Data received on port 1234 from port 1234 with length 4
... ...  Data received on port 1234 from port 1234 with length 4
... ...  Sending broadcast
... ...  Data received on port 1234 from port 1234 with length 4
... ...  Data received on port 1234 from port 1234 with length 4
... ...  Data received on port 1234 from port 1234 with length 4
```

Figure 4.8: Mote Data Sending and Receiving



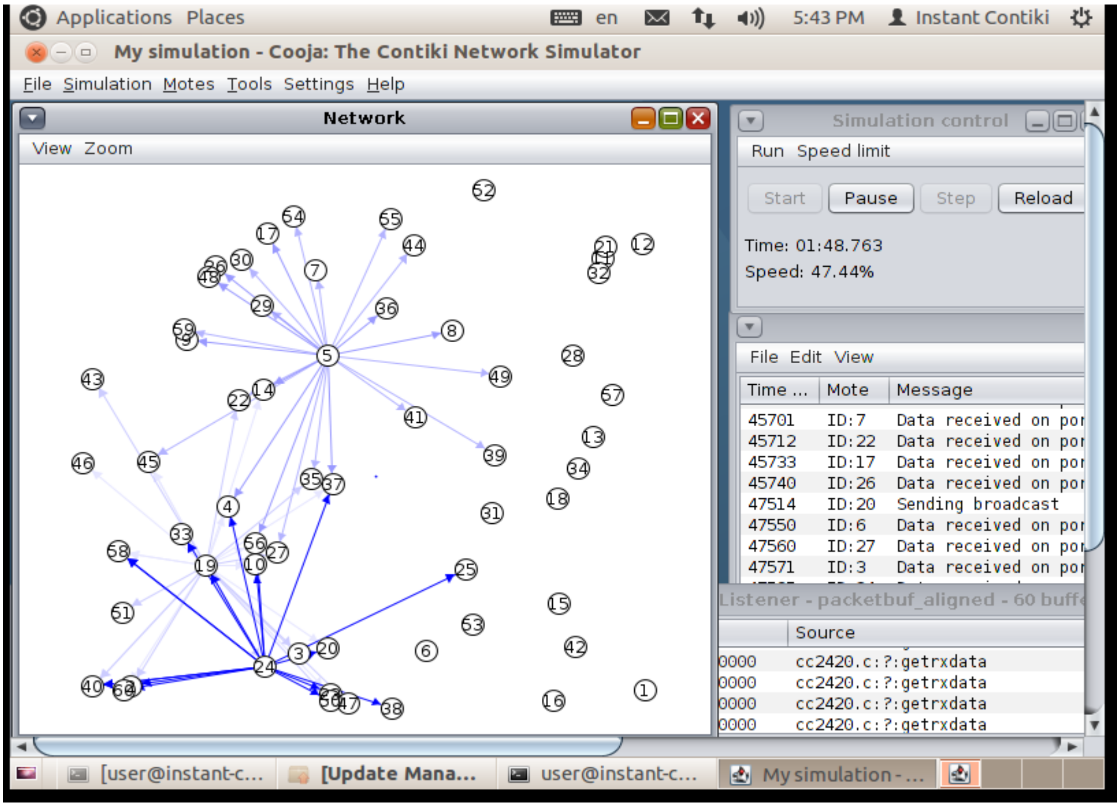
Mote: Sky 4
Time (ms): 79387.5
Radio HW was turned on

Figure 4.9: Timeline Representation



```
No.     Time        From  To   Data
330     01:24.615   29    14   70: 0x41C806CD ABFFFF1D 1D...
331     01:24.618   29    -    70: 0x41C806CD ABFFFF1D 1D...
332+1   01:24.622   29    9    70: 0x41C806CD ABFFFF1D 1D...
334+1   01:24.629   29    5    70: 0x41C806CD ABFFFF1D 1D...
336+1   01:24.636   29    4    70: 0x41C806CD ABFFFF1D 1D...
338+3   01:24.643   29    -    70: 0x41C806CD ABFFFF1D 1D...
342+1   01:24.658   29    30   70: 0x41C806CD ABFFFF1D 1D...
344     01:24.665   29    -    70: 0x41C806CD ABFFFF1D 1D...
```

Figure 4.10: Radio Message Analyze

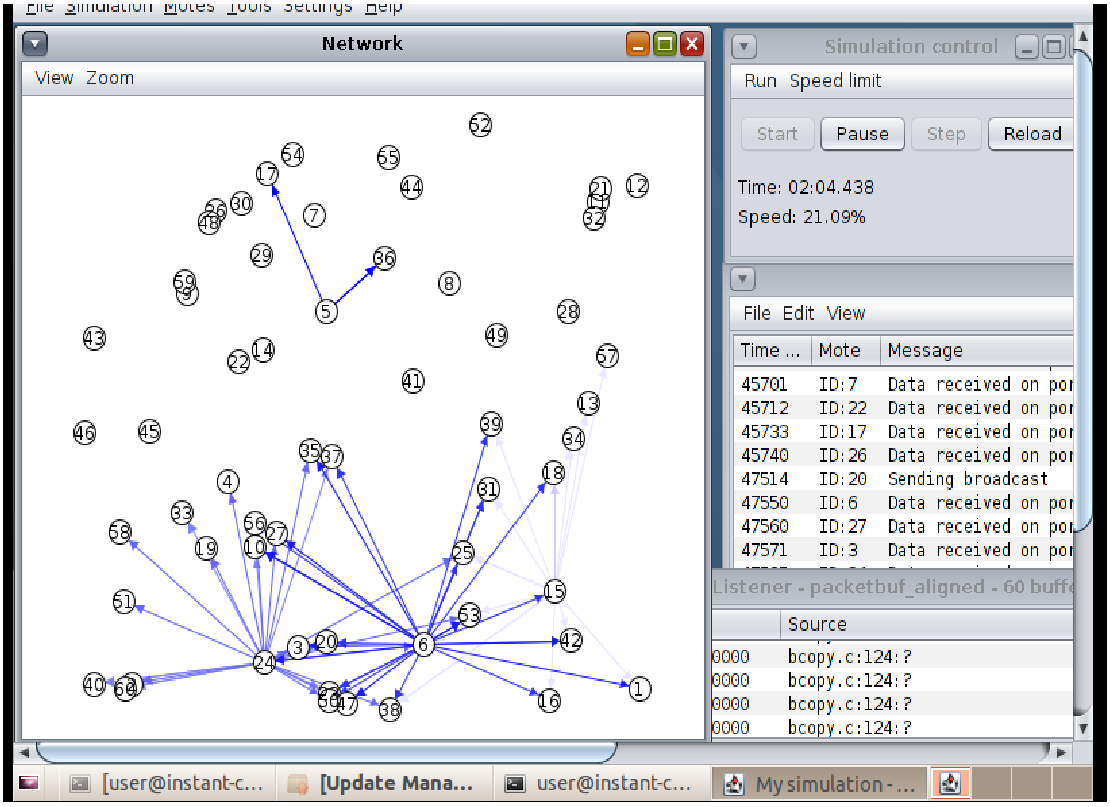


```
Time: 01:48.763
Speed: 47.44%
```

```
Time ...  Mote   Message
45701     ID:7   Data received on por
45712     ID:22  Data received on por
45733     ID:17  Data received on por
45740     ID:26  Data received on por
47514     ID:20  Sending broadcast
47550     ID:6   Data received on por
47560     ID:27  Data received on por
47571     ID:3   Data received on por
```

```
        Source
0000    cc2420.c:?:getrxdata
0000    cc2420.c:?:getrxdata
0000    cc2420.c:?:getrxdata
0000    cc2420.c:?:getrxdata
```

Figure 4.11: Communication Diagram-1

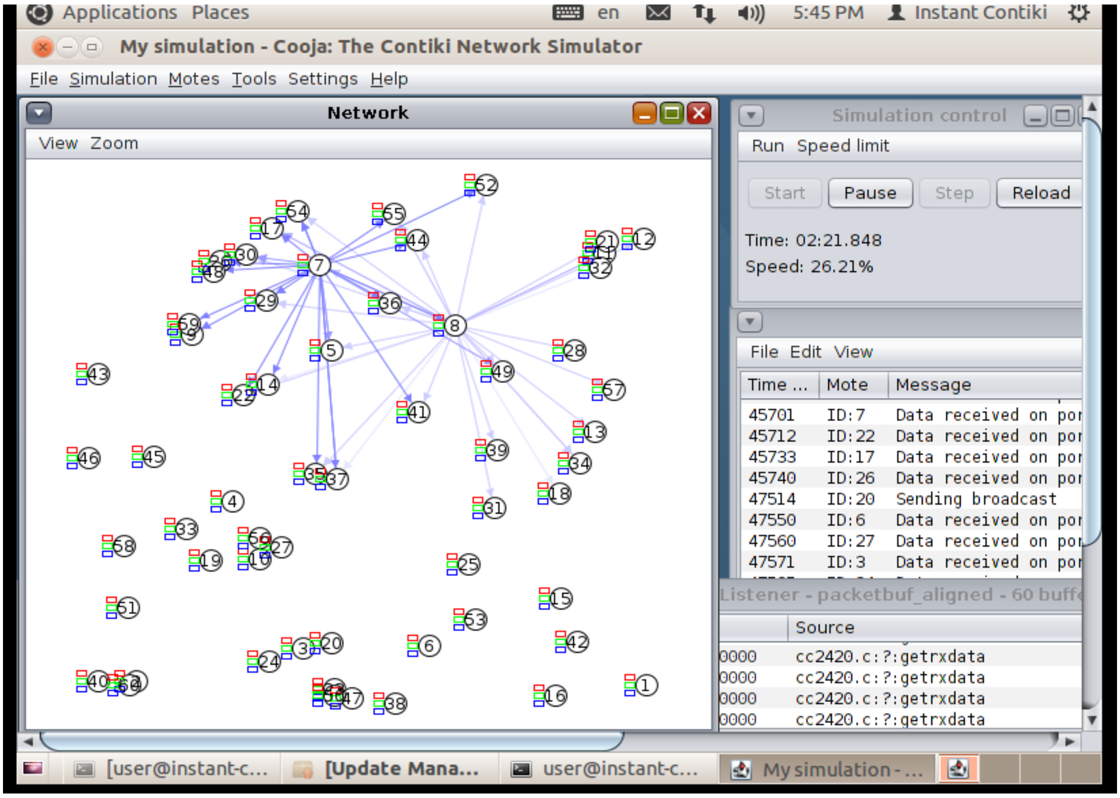


```
Time: 02:04.438
Speed: 21.09%
```

```
Time ...  Mote    Message
45701     ID:7    Data received on por
45712     ID:22   Data received on por
45733     ID:17   Data received on por
45740     ID:26   Data received on por
47514     ID:20   Sending broadcast
47550     ID:6    Data received on por
47560     ID:27   Data received on por
47571     ID:3    Data received on por
```

```
Listener - packetbuf_aligned - 60 buffe
          Source
0000      bcopy.c:124:?
0000      bcopy.c:124:?
0000      bcopy.c:124:?
0000      bcopy.c:124:?
```

Figure 4.12: Communication Diagram-2

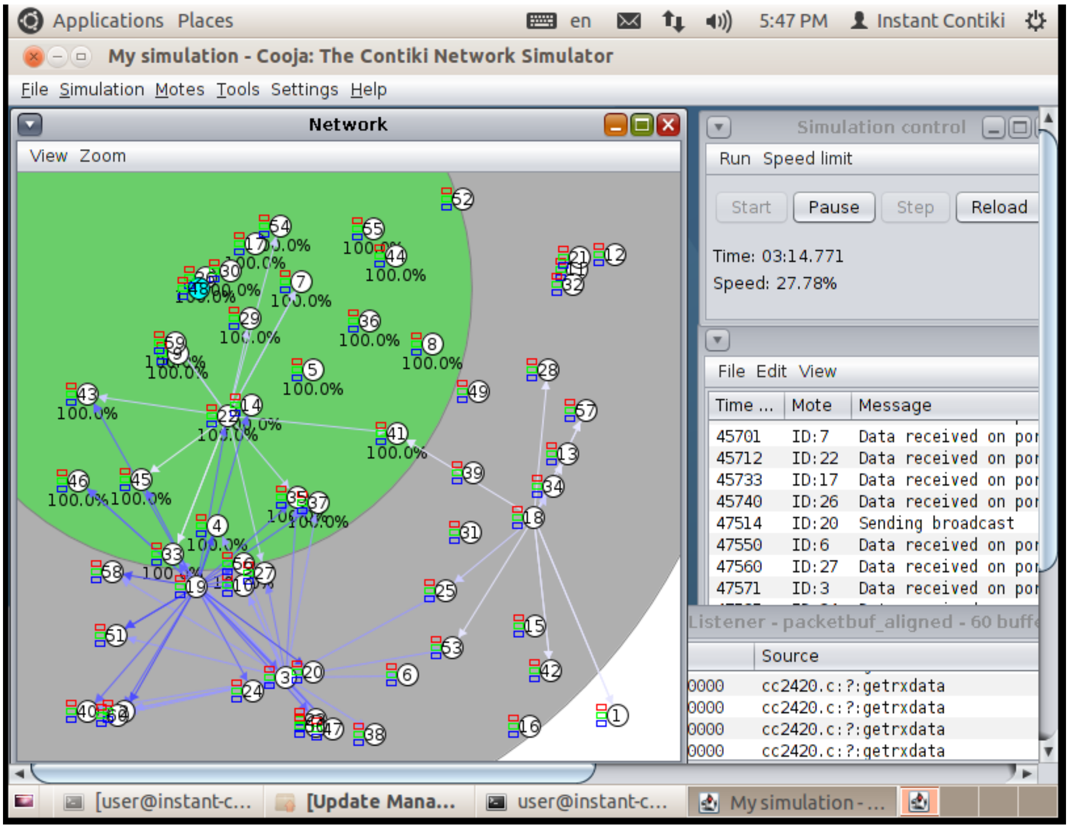


```
Time: 02:21.848
Speed: 26.21%
```

```
Time ...  Mote    Message
45701     ID:7    Data received on por
45712     ID:22   Data received on por
45733     ID:17   Data received on por
45740     ID:26   Data received on por
47514     ID:20   Sending broadcast
47550     ID:6    Data received on por
47560     ID:27   Data received on por
47571     ID:3    Data received on por
```

```
Listener - packetbuf_aligned - 60 buffe
          Source
0000      cc2420.c:?:getrxdata
0000      cc2420.c:?:getrxdata
0000      cc2420.c:?:getrxdata
0000      cc2420.c:?:getrxdata
```

Figure 4.13: Communication Diagram-3

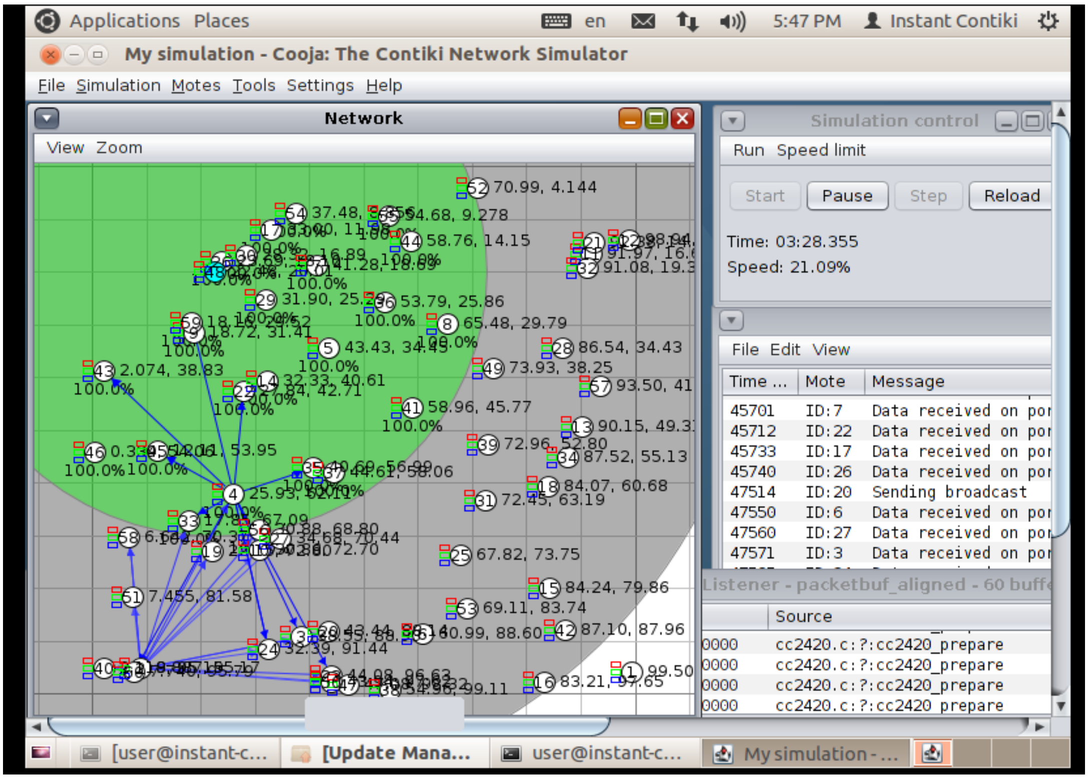


Time: 03:14.771
Speed: 27.78%

```
Time ...  Mote   Message
45701     ID:7   Data received on por
45712     ID:22  Data received on por
45733     ID:17  Data received on por
45740     ID:26  Data received on por
47514     ID:20  Sending broadcast
47550     ID:6   Data received on por
47560     ID:27  Data received on por
47571     ID:3   Data received on por
```

```
        Source
0000    cc2420.c:?:getrxdata
0000    cc2420.c:?:getrxdata
0000    cc2420.c:?:getrxdata
0000    cc2420.c:?:getrxdata
```

Figure 4.14: Network Radio Environment

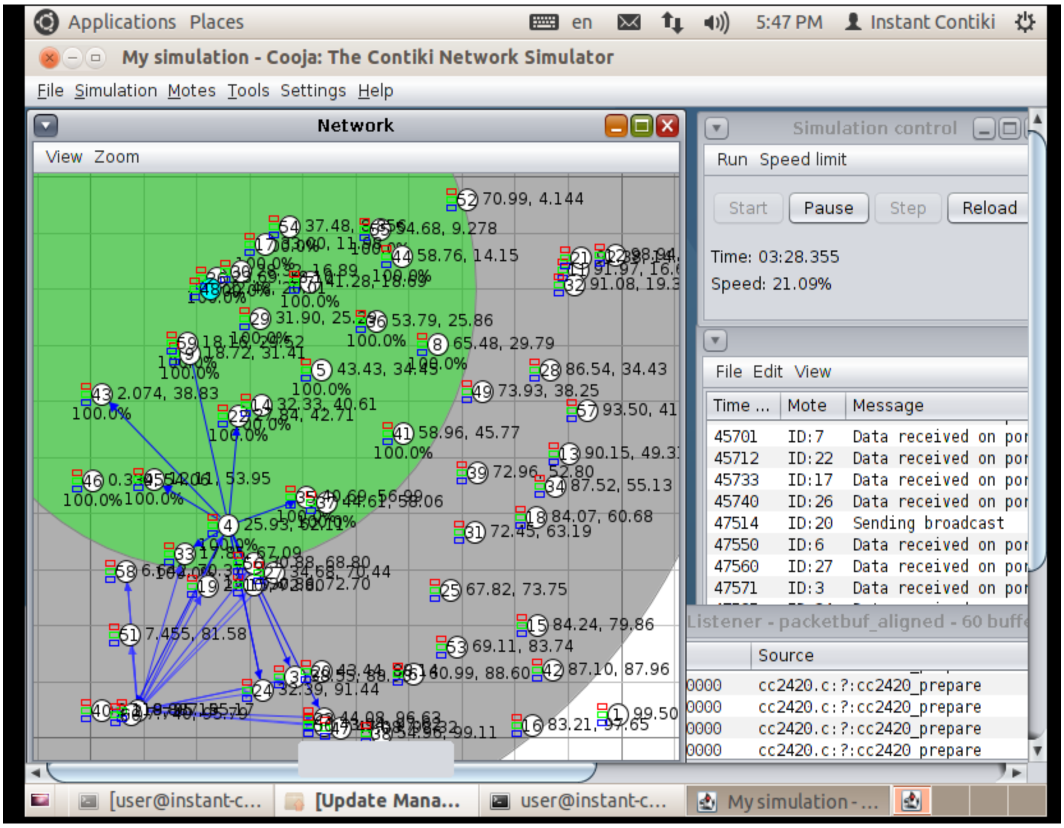


Time: 03:28.355
Speed: 21.09%

```
Time ...  Mote   Message
45701     ID:7   Data received on por
45712     ID:22  Data received on por
45733     ID:17  Data received on por
45740     ID:26  Data received on por
47514     ID:20  Sending broadcast
47550     ID:6   Data received on por
47560     ID:27  Data received on por
47571     ID:3   Data received on por
```

```
        Source
0000    cc2420.c:?:cc2420_prepare
0000    cc2420.c:?:cc2420_prepare
0000    cc2420.c:?:cc2420_prepare
0000    cc2420.c:?:cc2420_prepare
```

Figure 4.15: Network with Dynamic Nodes Position-1

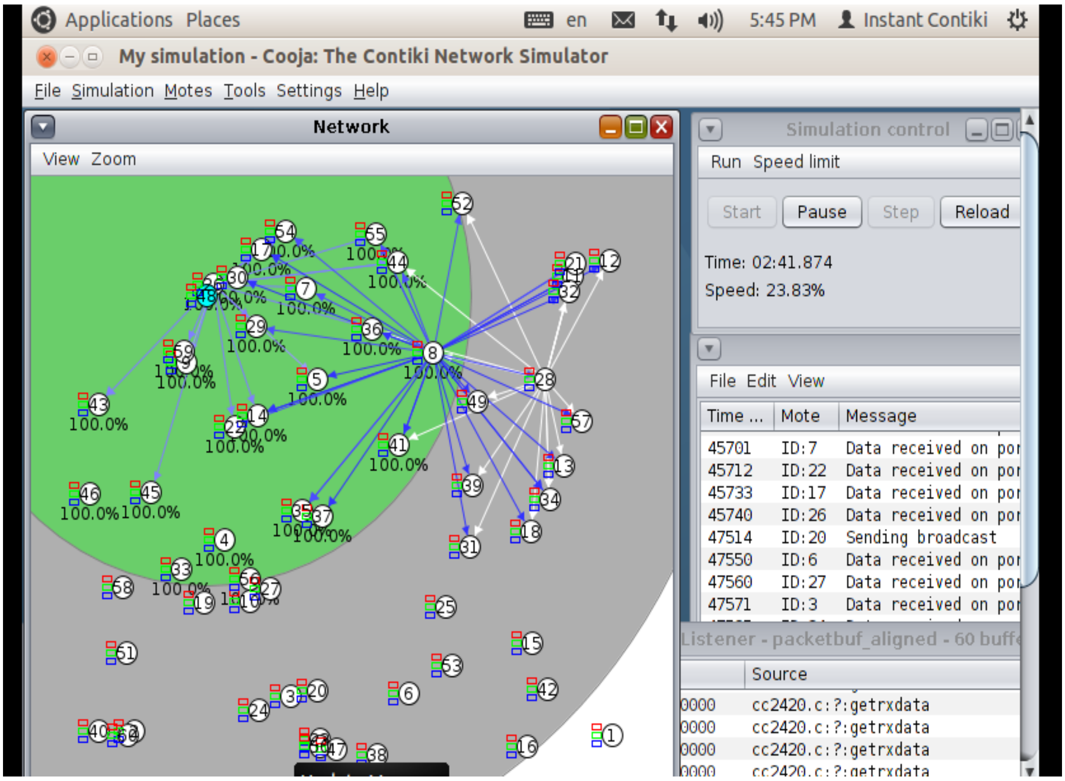


Time: 03:28.355
Speed: 21.09%

```
45701   ID:7    Data received on por
45712   ID:22   Data received on por
45733   ID:17   Data received on por
45740   ID:26   Data received on por
47514   ID:20   Sending broadcast
47550   ID:6    Data received on por
47560   ID:27   Data received on por
47571   ID:3    Data received on por
```

```
0000    cc2420.c:?:cc2420_prepare
0000    cc2420.c:?:cc2420_prepare
0000    cc2420.c:?:cc2420_prepare
0000    cc2420.c:?:cc2420_prepare
```

Figure 4.16: Network with Dynamic Nodes Position-2



Time: 02:41.874
Speed: 23.83%

```
45701   ID:7    Data received on por
45712   ID:22   Data received on por
45733   ID:17   Data received on por
45740   ID:26   Data received on por
47514   ID:20   Sending broadcast
47550   ID:6    Data received on por
47560   ID:27   Data received on por
47571   ID:3    Data received on por
```

```
0000    cc2420.c:?:getrxdata
0000    cc2420.c:?:getrxdata
0000    cc2420.c:?:getrxdata
0000    cc2420.c:?:getrxdata
```

Figure 4.17: Network with Dynamic Nodes Position-3

4.3 Performance Parameters

Following parameters are chosen for assessing the performance of security algorithms in this research work:

a) **Execution Time:** The time taken in total to generate a cipher text using a plain text is referred as execution time. This time leads to evaluate the speed of encryption. Execution time is measured in milliseconds (ms).

b) **Memory Requirements:** It is the number of bytes taken by the algorithm to save the temporary key and values generated during execution. Generally it includes RAM requirements. Memory requirements are measured in terms of number of bytes required by an algorithm.

c) **Throughput:** The encryption throughput is referred as the speed for encryption. It is the ratio of total size of varying plaintext with average time taken to encrypt all the plaintext. Throughput is measured in Kb/msec.

d) **Avalanche Effect:** With a small change in input if big change is noticed in the output bits, this is referred as avalanche effect. This property relates to diffusion of algorithm and reflects strength of the cryptographic algorithm. An algorithm having high avalanche effect is considered strong. Avalanche effect depends on hamming distance. Hamming distance is calculated as the sum obtained by performing bit by bit XOR.

e) **Energy Efficiency:** An algorithm is claimed to be energy efficient, if takes less number of CPU cycles for processing the plaintext.

4.4 Comparison of Existing Lightweight Algorithms and LDS

IoT can use internet only for connecting and establishing communication between things in the network. There is much more to further work upon in IoT like making decisions after communicating and that too in real time. So, architecture of internet cannot be directly employed for IoT.

CoAP was earlier using the security of IPSec and DTLS. The predefined security mechanisms are vulnerable to aforementioned attacks. So, cryptography algorithms can be incorporated in them. Cryptography algorithms can be symmetric and asymmetric.

Symmetric algorithm uses a single private key for communication. Sender and receiver share same key for communication. Symmetric key assures confidentiality and integrity of data, but do not guarantee authentication. Advantage of symmetric is less number of keys required with less key size. Disadvantage is secure key distribution among both the parties, and it does not authenticate the sender.

Asymmetric uses pair of public and private key for communication. Asymmetric assures confidentiality, integrity, and authentication. For confidentiality and integrity sender encrypts the data using public key of receiver that can be only decrypted by private key of receiver. To assure authentication, data is encrypted by private key of sender and receiver confirms it by decrypting it with public key of sender. Advantage of Asymmetric cryptography is it supports all security services, but disadvantage is the large size of key which will increase the complexity of algorithm.

This research work compares the existing lightweight algorithms SPECK, SIMON, FANTOMAS, TWINE and the proposed LDS by executing them on COOJA. Performance of these algorithms is compared on the basis of their execution time, memory requirements, throughput, avalanche effect, and energy efficiency.

a) **Execution Time:** Performance comparison of SPECK, SIMON, FANTOMAS, TWINE, and LDS is made on basis of their execution time. Algorithm bearing the high encryption speed for converting plaintext to cipher text will take less execution time.

   Figure 4.18 represents the comparison of execution time of SPECK, SIMON, FANTOMAS, TWINE, and LDS. The execution time of LDS is 11 % less as compared to SPECK, 16 % less as compared to SIMON, 12 % less as compared to FANTOMAS, and 54 % less as compared to TWINE. For an algorithm to be efficient, its execution time should be less. Less execution time will take up less resources like power of IoT devices.

   Therefore, it can be concluded that LDS has least execution time over other security algorithms.

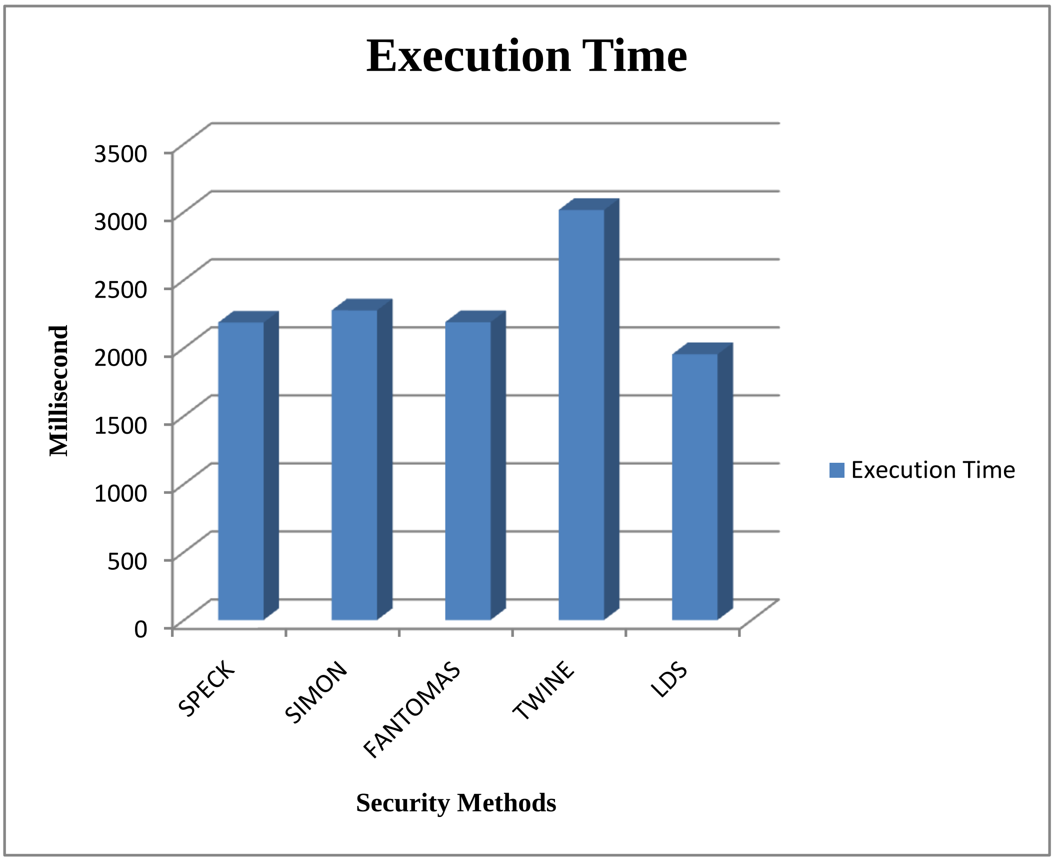


Figure 4.18: Comparison of Existing Security Algorithms and LDS on the basis of Execution Time

b) **Memory Requirements:** Performance comparison of SPECK, SIMON, FANTOMAS, TWINE and LDS is made on basis of their memory requirements. Memory can be used to store the code size of application. Sensing data and other temporary data is also stored in memory. The algorithm bearing the fewer memory requirements will execute faster.

Figure 4.19 represents the comparison of memory requirements of SPECK, SIMON, FANTOMAS, TWINE, and LDS. The memory requirements of LDS is 11% less as compared to SPECK, 26 % less as compared to SIMON, 22 % less as compared to FANTOMAS, and 32 % less as compared to TWINE. For an algorithm to be efficient, its memory requirements should be less.

Therefore, it can be concluded that LDS has least memory requirements over other security algorithms.

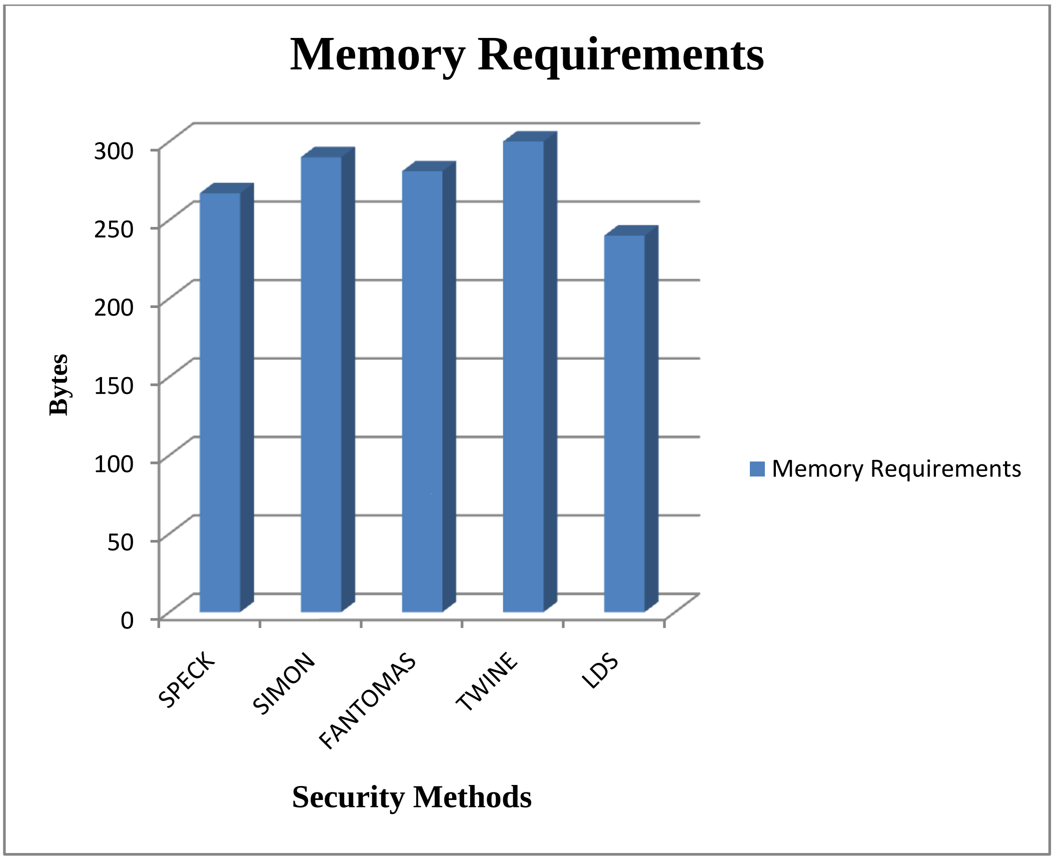


Figure 4.19: Comparison of Existing Security Algorithms and LDS on the basis of Memory Requirement

c) **Throughput:** Performance comparison of SPECK, SIMON, FANTOMAS, TWINE, and LDS is made on basis of their throughput. Throughput is determined by the number of bytes of plaintext encrypted over the period of time. The algorithm with fast encryption speed will have high throughput.

Figure 4.20 represents the comparison of throughput of SPECK, SIMON, FANTOMAS, TWINE, and LDS. The throughput of LDS is 11 % more as compared SEPCK, 20 % more as compared to SIMON, 16 % more as compared to FANTOMAS, and 26 % more as compared to TWINE. For an algorithm to be efficient, its throughput should be high. If throughput is high, more number of bytes of plaintext can be encrypted in a particular span of time leading to fast operation of algorithm.

Therefore, it can be concluded that LDS has high throughput over other security algorithms.

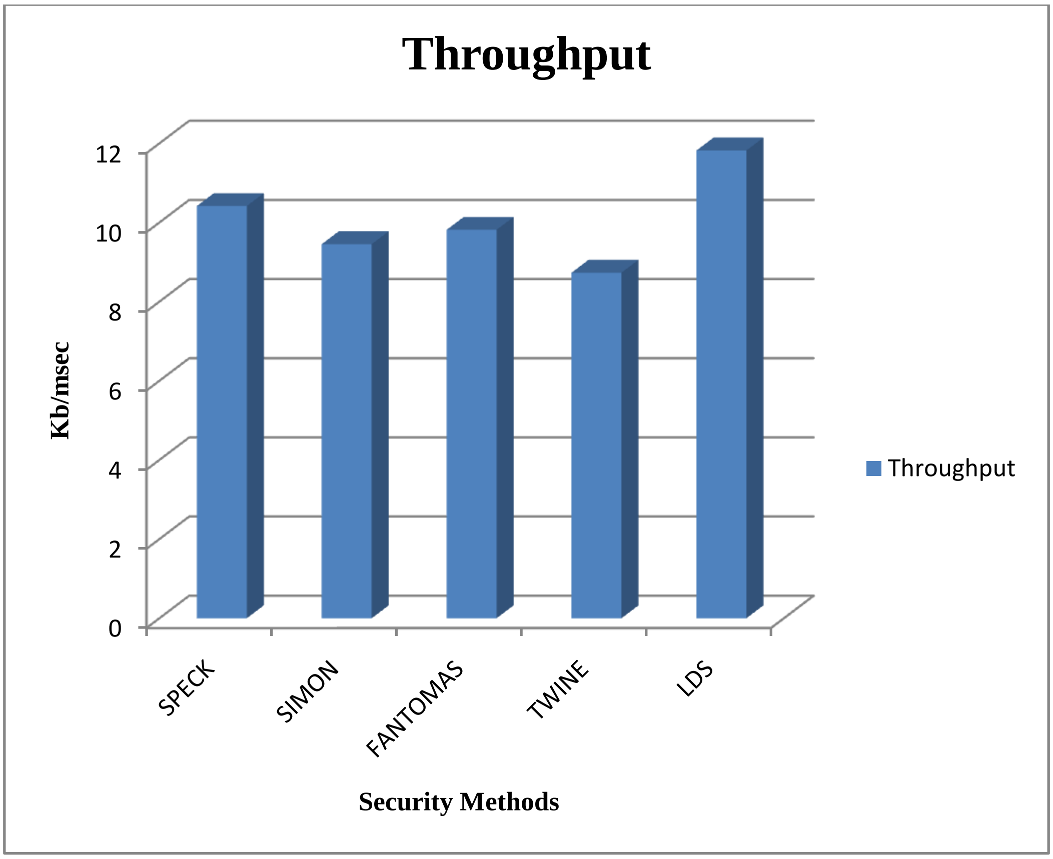


Figure 4.20: Comparison of Existing Security Algorithms and LDS on the basis of Throughput

d) Avalanche Effect: Performance comparison of SPECK, SIMON, FANTOMAS, TWINE, and LDS is made on basis of the diffusion made in output by changing 1 single input bit. Avalanche effect is dependent on the hamming distance. The algorithm having high avalanche effect is stronger.

Figure 4.21 represents the percentage comparison of avalanche effect on SPECK, SIMON, FANTOMAS, TWINE, and LDS. The diffusion in LDS is 8 % more as compared to SPECK, 17 % more as compared to SIMON, 26 % more as compared to FANTOMAS, and 51 % more as compared to TWINE. For an algorithm to be more secure, it should have high avalanche effect. Therefore, if any attacker tries to modify the content of plaintext changing a single bit in input may affect multiple bits of output.

Therefore, it can be concluded that LDS has high diffusion property over other security algorithms.

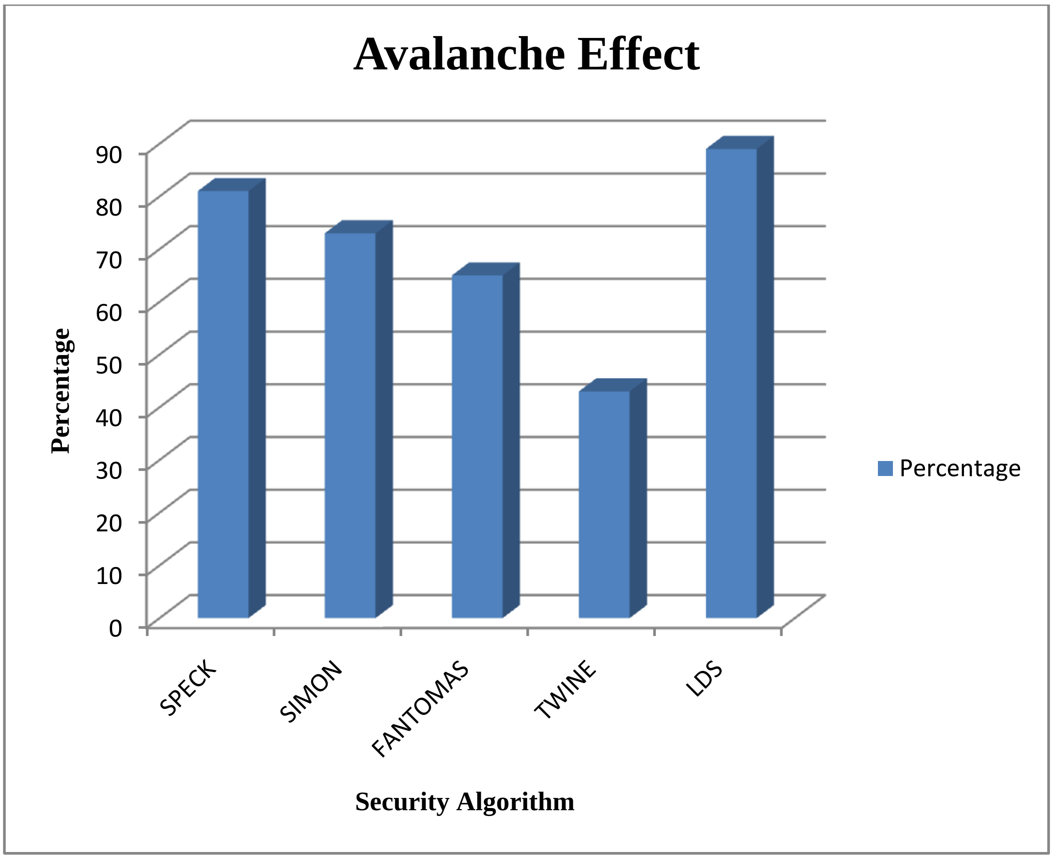


Figure 4.21: Comparison of Existing Security Algorithms and LDS on the basis of Avalanche Effect

e) **Energy Efficiency:** Performance comparison of SPECK, SIMON, FANTOMAS, TWINE, and LDS is made on basis of how much energy efficient they are. The number of CPU cycles determines the energy efficiency of the algorithm. The algorithm executing in less CPU cycles, consumes less power and hence is more energy efficient.

Figure 4.22 represents the comparison of CPU cycles taken by SPECK, SIMON, FANTOMAS, TWINE, and LDS. The number of CPU cycles for LDS is 6 % less as compared to SPECK, 24 % less as compared to SIMON, 22 % less as compared to FANTOMAS, and 20 % less as compared to TWINE.

Therefore, it can be concluded that LDS requires less number of CPU cycles over other security algorithms.

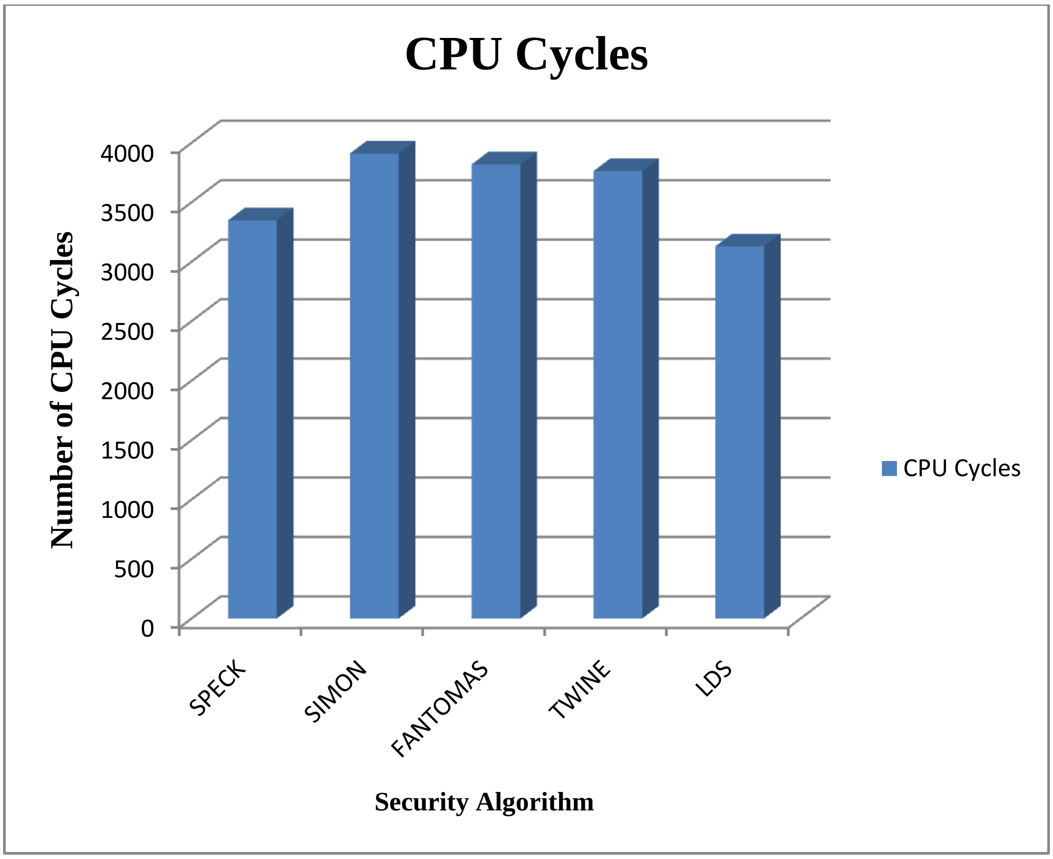


Figure 4.22: Comparison of Existing Security Algorithms and LDS on the basis of Energy Efficiency

Lightweight block cipher algorithms like SPECK, SIMON, FANTOMAS, TWINE, and LDS are compared on the basis of their software implementation using C language in COOJA simulator. All the algorithms were executed on data collected from the motes in the scenario. A pair of sender and receiver was taken and security was imposed on data in the transit.

Various parameters taken to compare the algorithms are code length, memory requirements, execution time, throughput, avalanche effect, energy efficiency (CPU Cycles). For an algorithm to perform in an efficient manner it should posses high throughput, less memory requirements, less execution time, number of CPU cycles consumed for its execution should be less.

On the basis of these performance parameters, comparison is made among the existing algorithms SPECK, SIMON, FANTOMAS, TWINE and proposed LDS and is concluded through Table 4.2 as presented below.

Table 4.2: Performance Comparison of Existing Lightweight Algorithms and LDS

Block Cipher	Code length	Memory Requirements (Bytes)	Execu-tion Time (msec)	Throughput (kb/msec)	Avalanche Effect (%)	CPU Cycles
Speck	1630	267	2186	10.416	81	3348
Simon	2200	290	2276	9	73	3910
Fantomas	5600	281	2189	9.815	65	3819
Twine	4203	300	3014	8.732	43	3762
LDS	1712	240	1953	11.819	89	3129

Both SIMON and SPECK are vulnerable to the differential attack but SPECK gives high throughput as compared to SIMON. On the other hand, even though FANTOMAS does not require tables S-Boxes that are expensive to mask but L-Boxes computation cost is more. At last, TWINE takes more execution time and has less throughput and is also susceptible to linear attack with zero co-relation.

From the above comparison table it can be deduced that taking into consideration the different evaluation parameters like execution time, memory required, energy efficiency, throughput, avalanche effect, **LDS block cipher outperforms all other existing security mechanisms,** bearing less execution time, less memory requirements, high energy efficiency, and more avalanche effect.

4.5 Performance Comparison of Proposed LDS Algorithm and SPECK

In Chapter 3, a HLSF is designed with a phase 3 dedicated towards proposing a **LDS Algorithm** to offer data confidentiality and integrity. This section compares the better performing algorithm SPECK among the existing security algorithms with the proposed LDS algorithm. Same parameters as stated above in section 4.3, memory requirements, energy efficiency, execution time and throughput are chosen to evaluate their performance. Two variations of Key size/Block Size are opted for both SPECK and LDS. That is, SPECK 64/96 having 64 as block size and 96 as key size is compared with LDS 64/96. Further SPECK 64/128 is compared with LDS 64/128.

a) **Execution Time:** Performance comparison of SPECK and LDS is made on basis of their execution time. Algorithm bearing the high encryption speed for converting plaintext to cipher text will take less execution time.

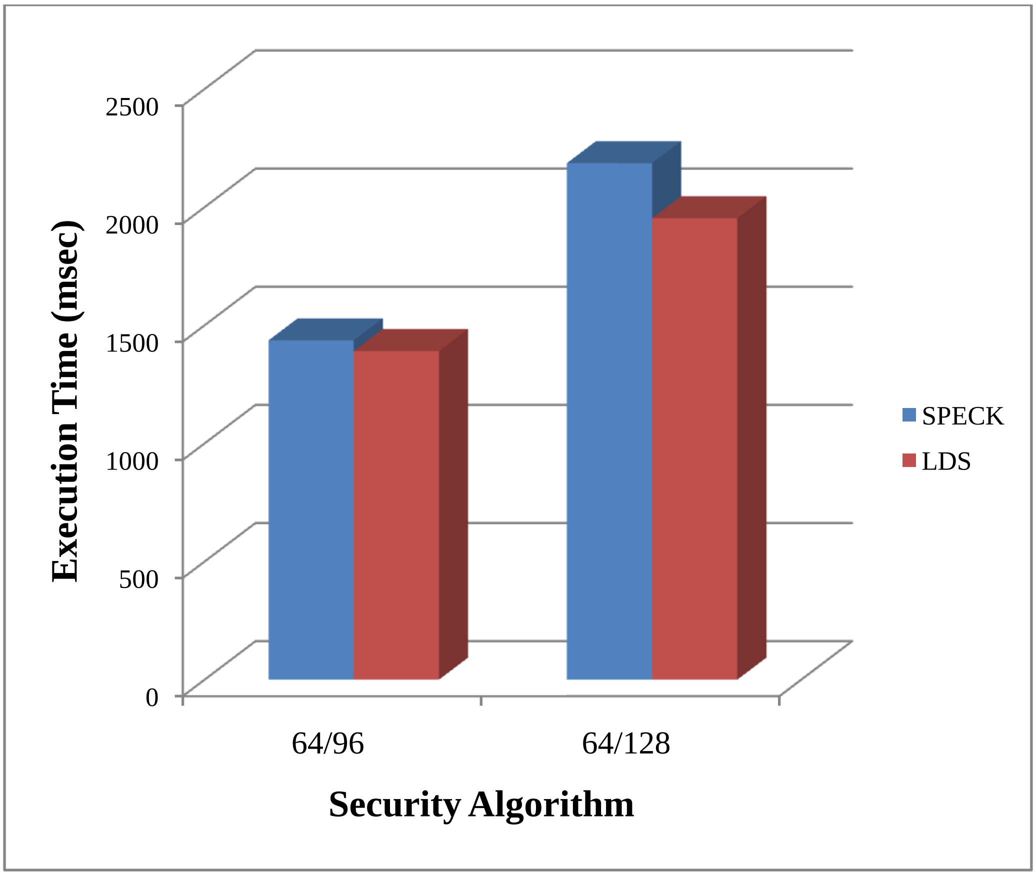


Figure 4.23: Comparison of SPECK and LDS on the basis of Execution Time

Figure 4.23 represents the comparison of memory requirements of SPECK 64/96 with LDS 64/96 and SPECK 64/128 with LDS 64/128. The execution time of LDS 64/96 is 3% less as compared to SPECK 64/96. The execution time of LDS 64/128 is 11 % less as compared to SPECK 64/128. Therefore, it can be concluded that LDS has least execution time over SPECK considering both the variants.

b) **Memory Requirements:** Performance comparison of SPECK and LDS is made on basis of their memory requirements. The algorithm bearing the fewer memory requirements will execute faster.

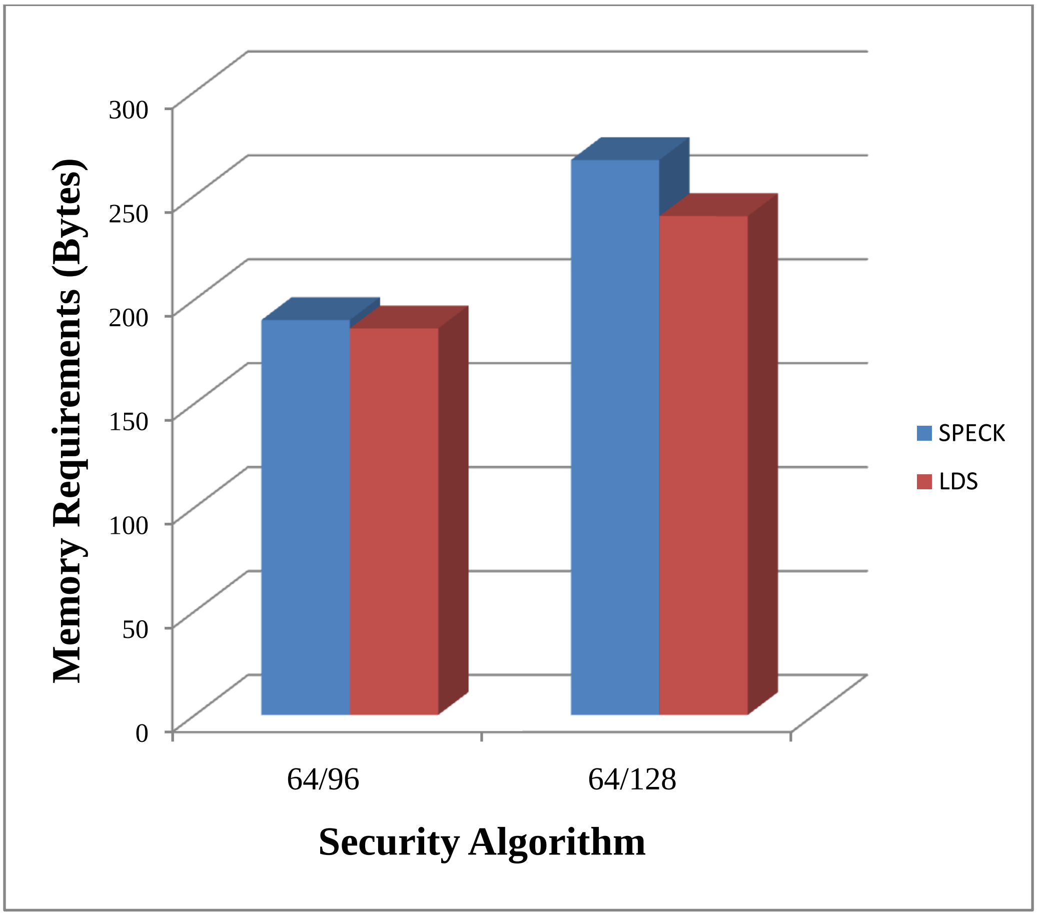


Figure 4.24: Comparison of SPECK and LDS on the basis of Memory Requirement

Figure 4.24 represents the comparison of memory requirements of SPECK 64/96 with LDS 64/96 and SPECK 64/128 with LDS 64/128. The memory requirements of LDS 64/96 are 2 % less as compared to SPECK 64/96. The memory requirements of LDS 64/128 are 11 % less as compared to SPECK 64/128. Therefore, it can be concluded that LDS has less memory requirements over SPECK considering both the variants.

c) **Throughput:** Performance comparison of SPECK and LDS is made on basis of their throughput. The algorithm with fast encryption speed will have high throughput.

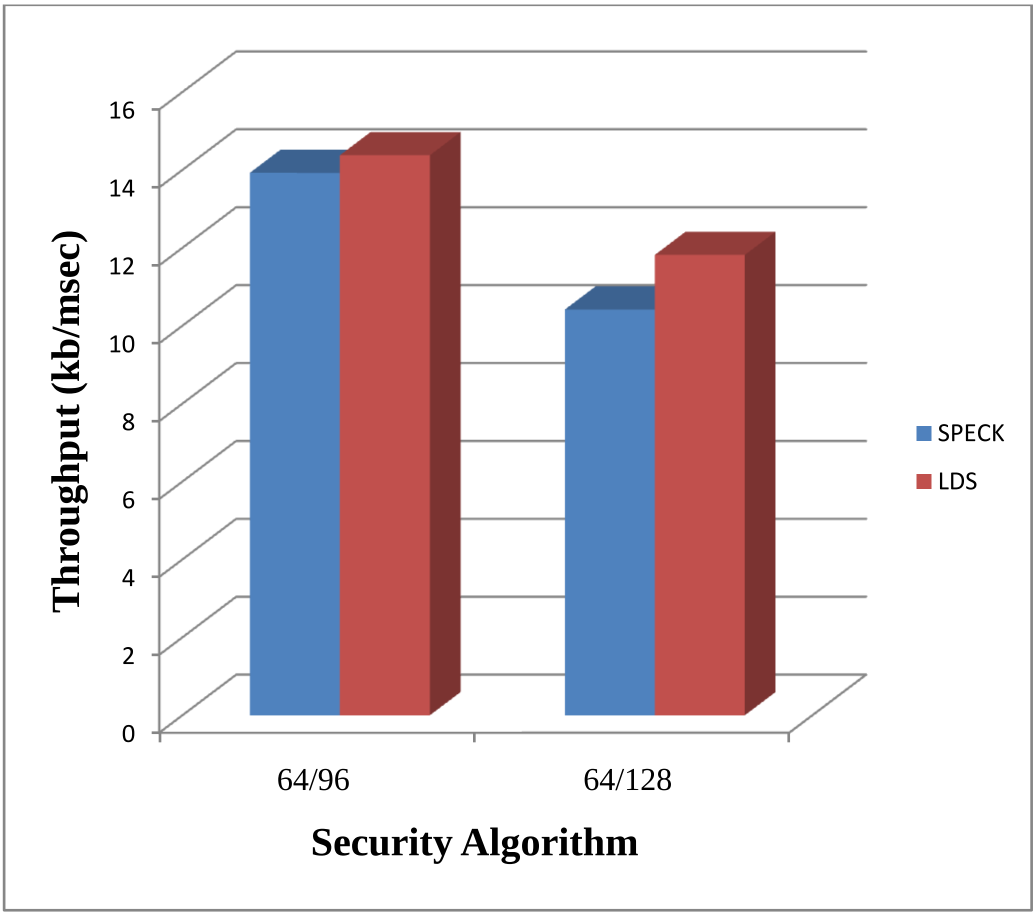


Figure 4.25: Comparison of SPECK and LDS on the basis of Throughput

Figure 4.25 represents the comparison of throughput of SPECK 64/96 with LDS 64/96 and SPECK 64/128 with LDS 64/128. The throughput of LDS 64/96 is 3% more as compared to SPECK 64/96. The throughput of LDS 64/128 is 12 % more as compared to SPECK 64/128. Therefore, it can be concluded that LDS gives high throughput over SPECK considering both the variants.

d) Avalanche Effect: Performance comparison of SPECK and LDS is made on basis of how much diffusion property they contain. If hamming distance is more, avalanche effect will be high. The algorithm where diffusion is more, that is, hamming distance is more, avalanche effect will be more, hence is the strongest algorithm.

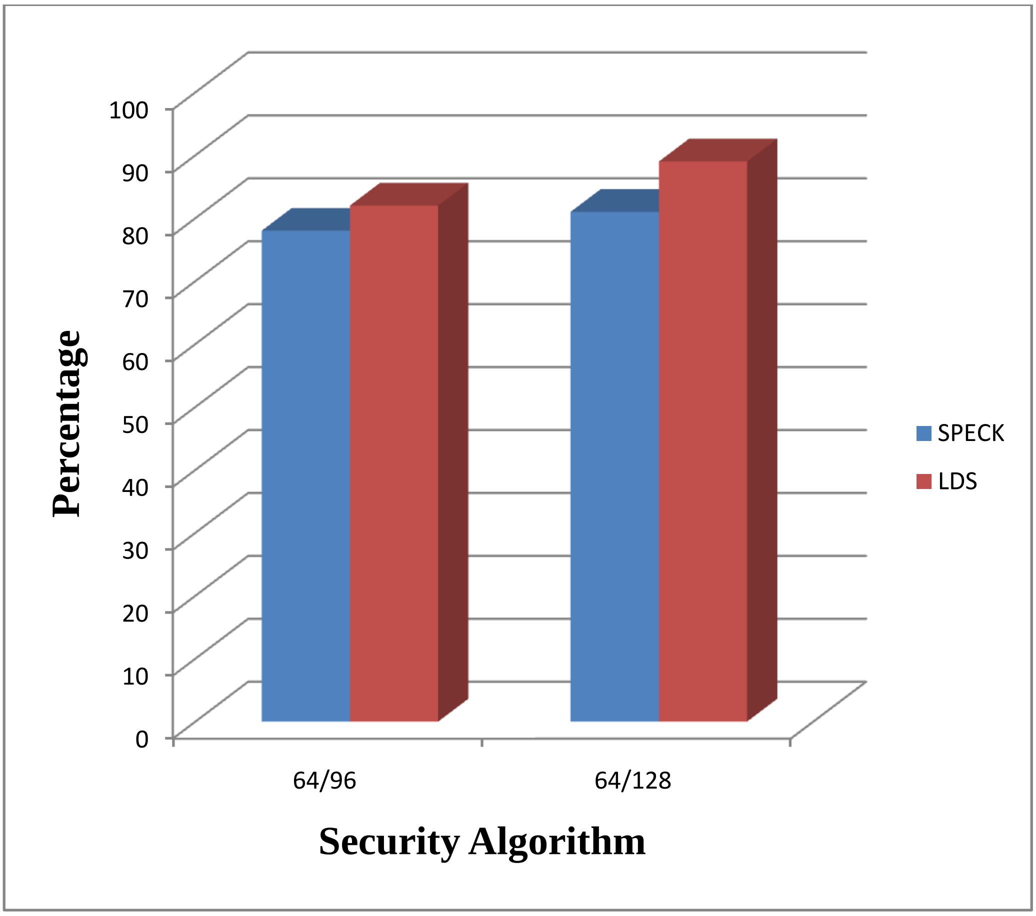


Figure 4.26: Comparison of SPECK and LDS on the basis of Avalanche Effect

Figure 4.26 represents the percentage comparison of avalanche effect on SPECK 64/96 with LDS 64/96 and SPECK 64/128 with LDS 64/128. The avalanche effect of LDS 64/96 is 5% more as compared to SPECK 64/96. The avalanche effect of LDS 64/128 is 9 % more as compared to SPECK 64/128.

Therefore, it can be concluded that LDS is stronger over SPECK considering both the variants.

e) **Energy Efficiency:** Performance comparison of SPECK and LDS is made on basis of how much energy efficient they are. The number of CPU cycles determines the energy efficiency of the algorithm. The algorithm executing in less CPU cycles, consumes less power and hence is more energy efficient.

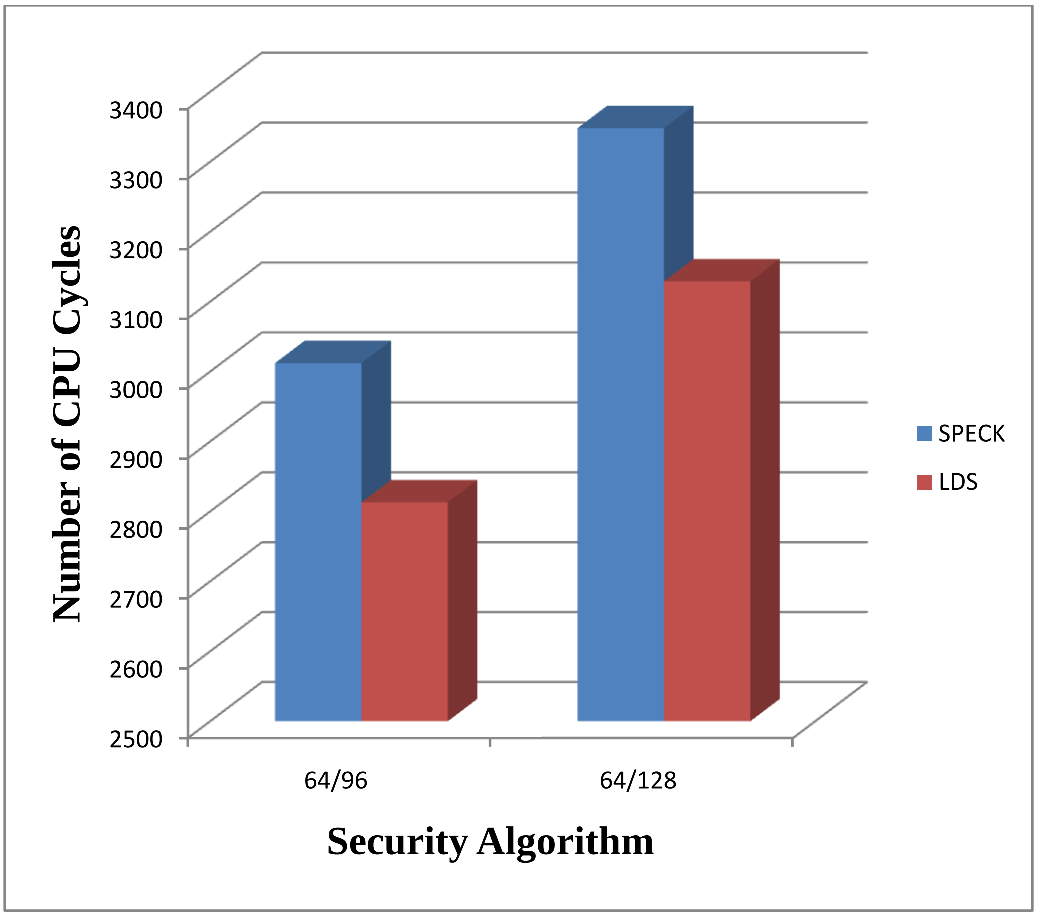


Figure 4.27: Comparison of SPECK and LDS on the basis of Energy Efficiency

Figure 4.27 represents the comparison of number of CPU cycles of SPECK 64/96 with LDS 64/96 and SPECK 64/128 with LDS 64/128. The number of CPU cycles of LDS 64/96 is 7% less as compared to SPECK 64/96. The number of CPU cycles of LDS 64/128 is 6% less as compared to SPECK 64/128. Therefore, it can be concluded that LDS is more energy efficient as compared to SPECK considering both the variants.

Proposed cryptographic algorithm and SPECK are compared on the basis of code length, block size, key size, memory requirements, execution time, throughput, avalanche effect, energy efficiency (CPU Cycles) as shown in Table 4.3 below

Table 4.3: Performance Comparison of LDS and SPECK

Algorithm	**Block Size**	**Key Size**	**Code length**	**Memory Requirements (Bytes)**	**Execution Time (msec)**	**Throughput (kb/msec)**	**Avalanche Effect (%)**	**CPU Cycles**
SPECK	64	96	1630	190	1436	13.927	78	3012
LDS			1712	186	1391	14.378	82	2813
SPECK	64	128	1630	267	2186	10.416	81	3348
LDS			1712	240	1953	11.819	89	3129

4.6 Summary

This chapter does the performance comparison of existing lightweight cryptographic algorithms available in literature. For evaluating the performance, COOJA simulator is used that works over CONTIKI OS. Code Length, Memory requirements, Execution time, Throughput, Avalanche effect, Energy efficiency (CPU Cycles) are evaluated for the existing cryptographic algorithms and LDS. Results show that LDS works well over other solutions. Later, a performance comparison is made among proposed LDS and SPECK with two variations 64/96 and 64/128. Results prove that LDS is better in performance in comparison with SPECK considering both the variants. Therefore, it can be concluded from this chapter LDS has less memory requirements, less execution time, high throughput, more avalanche effect, and is more energy efficient.

CHAPTER 5

SECURITY ANALYSIS OF PROPOSED SECURITY ALGORITHM AND SPECK IN IOT

Cryptography methods are aimed at offering communications that are secure. In order to find out the effectiveness of a cryptographic solution, cryptanalysis is performed to find vulnerabilities and the attacks possible on a cryptographic algorithm. This chapter exposes the vulnerabilities of SPECK block cipher and proposed LDS algorithm. Based on the results the attack complexity is determined for them.

5.1 Introduction

Cryptanalysis works on the technique of gathering the information sufficient enough to guess the key by knowing the cipher text and the other details of the cryptographic algorithm. In this research work computational security is evaluated for SPECK and LDS. Computational security refers to exposing the cipher text for various attacks. If a cryptographic algorithm requires say “m” amount of computation to extract the information from cipher, it can be said that cipher has security amount of “m”.

An algorithm is fully secure if the attacker is not able to compute within “m” amount. But this kind of security is infeasible in practical life, except few techniques like one time pad, where a key is used for only one time. The cipher can be attacked to fetch either the key but that will not be sufficient, in other hand attacker will try to get the plain text directly from the behavior of cipher text.

A cipher or cryptographic algorithm is called as fully broken, if by knowing the cipher text either the attacker is able to get the key completely and can generate the further plaintext, or by knowing the partial key the attacker is able to get the information about the plaintext based on the cipher text- plain text combinations. There are different attacks possible as identified in literature on cryptographic algorithms. Each attack works on a different mechanism to break the security of the algorithm.

5.2 Attacks on Cryptographic Algorithms

Number of attacks tends to find the key in order to exploit the behavior of cryptographic algorithm. Attacks that are not able to find the key will find out a unique way of revealing the other behavior of algorithm that can indirectly fetch the key for attack. The attacks possible on cryptographic algorithms are summarized below:

- **Brute Force Attack:** In this attack the cipher is tested by all the possible assumption of keys made by the attacker. This attack is not efficient as it is becomes very complex when key size is more. Therefore, from security perspective, to design a secure algorithm the key size should not be less as it can be easily brute forced by the attacker.
- **Cipher text only attacks:** In this attack, only thing attacker knows is few cipher texts. Therefore, this is the most impactful attack and can be possible on the weakest cryptographic algorithm. If a algorithm is vulnerable to this attack, it is the one of the weak solutions to the system.
- **Known Plaintext Attack:** Attackers in this scenario, have access to some pairs of cipher text and the corresponding plain text pair. This pair can be retrieved by the attacker by eaves dropping the system through a communication line. Therefore, using this pair, attacker will work on finding the key used in the algorithm. **Linear cryptanalysis** is one of the examples of the known plain text attack.
- **Chosen Plaintext Attack:** In this attack, attacker queries the cipher text depending upon the set of chosen plain texts. Examples of this attack are, rotational cryptanalysis, rotational-XOR cryptanalysis, and **differential cryptanalysis**.
- **Chosen Cipher text Attack:** This attack is exactly reverse of the above chosen plain text attack. Here, the attacker is able to get the plaintext in decrypted form by choosing the corresponding cipher text.

Out of the above stated attacks, SPECK is vulnerable to linear and differential attacks. Next section performs differential cryptanalysis of SPECK and proposed LDS.

5.3 Finding Differential Probability

In 1991, DES was attacked first using the differential cryptanalysis [142]. XOR function when used in the encryption function of a algorithm, propagates differences in input as well as output of function. Let us say, the encryption function of algorithm is represented as F_k, where k is the key. Considering two sets of plaintext (P_1 and P_2) and cipher text (C_1 and C_2) generated from the equation10 and 11 as:

$$C1 = Fk\,(P1) \quad (10)$$

$$C2 = Fk\,(P2) \quad (11)$$

Differential cryptanalysis works on calculating two types of differences, the input difference and the output difference. Input difference (D_i) is the difference between the pair of inputs P_1 and P_2 and is calculated as $D_i = P_1 \oplus P_2$. Output difference (D_o) is the difference between the pair of outputs C_1 and C_2 and is calculated as $D_o = C_1 \oplus C_2$. The pair of input and the output difference (D_i, D_o) is called as the differential. Probability that D_i difference in input is going to produce D_o difference in output is referred as ***Differential Probability***, that is , P(D_i->D_o) and is represented through Figure 5.1

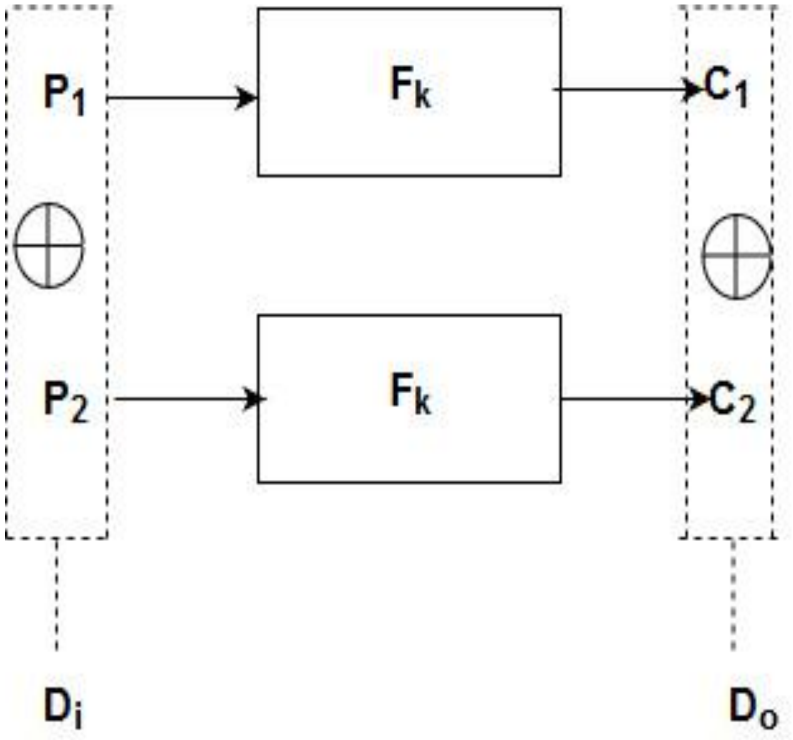


Figure 5.1: Differential D_i And D_o Representing Input and Output Differences

If such D_i and D_o exist with differential probability more than 0.5, it means differential attack can be mounted on that cipher.

5.4 Differential Cryptanalysis Algorithm

Differential cryptanalysis is a chosen plain text attack. The main concern of differential attack is to discover the key for different rounds of operation. Earlier, differential attack used counting as the technique to discover the key [143]**.** Counting technique makes use of statistical techniques to recover the keys but in chunks. Input pairs are analyzed to discover the keys. The collected chunks of keys may be correct or wrong. Therefore, a strong filtering mechanism is required to chop out the wrong chunks generated.

The key recovery process of differential cryptanalysis to find the master key is divided into two phases; one is Gathering and second is Guessing the key. The initial difference between plaintext is denoted by D_i. Later, if on this plaintext pair encryption is applied for r-1 rounds, then after r-1 rounds the difference in output is denoted as D_o.

Phase 1: Gathering Phase: This phase is used to generate the cipher text for r-1 rounds with a high differential probability.

Step 1: Choose two input pairs P_1 and P_2, where P_2= P1 $\oplus D_i$

Step 2: If total round of a block cipher are r, then encryption function is applied for r-1 rounds to find the output pairs C_1 and $C_{2.}$ Where C2= C1$\oplus D_{o.}$

Step 3: For the rth round apply the encryption function on the chosen plaintext P_1 and P_2 and obtain the corresponding output pair Y_1 and Y_2. Therefore, Y_1= E(P_1) and Y_2= E(P_2).

Phase 2: Guessing the Key: Brute force technique is applied for finding the master key by executing the following steps:

Step 1: Take a matrix for K_r key of the rth round by considering the every possible combination and by initializing counter of each key as 0.

K_r	K_r^1	K_r^2	K_r^3	K_r^i	K_r^{n-1}	K_r^n
	0	0	0	0	0	0

Here K_r^i represents the ith possible combination of key and n represents the number of bits in key as per key schedule.

Step 2: Using every possible combination of key, for example, K_r^i, apply the inverse encryption function as shown in equation 12 and 13 to find

$$F_r^{-1}(Y_1, K_r^i) = C_1 \quad (12)$$

$$F_r^{-1}(Y_2, K_r^i) = C_2 \quad (13)$$

Step 3: Calculate $C_1 \oplus C_2$, if it comes out to be D_0, that is the probable output difference, then increase the counter of that particular K_r^i by 1.

K_r	K_r^1	K_r^2	K_r^3	K_r^i	K_r^{n-1}	K_r^n
	0	0	0	1	0	0

Step 4: Repeat step 2 until all the possible key combinations are checked. The key with the highest value will be the actual rth round key.

5.5 Differential Cryptanalysis of SPECK

SPECK belongs to the family of Feistel network based on ARX structure. Here the input is processed as two words one termed as left and other as right. In each round, the left rotation is performed at the left side by let us x bits and the right rotation is performed at the right side by let us say y bits as shown in Figure 5.2 . The value of x and y is generally used as 8 and 3 in implementation of all most all versions of SPECK.

The key schedule in SPECK makes use of a separate round function that helps in generating round keys K_i for each round of operation. Therefore for SPECK 2n/mn, with n bit word and master key of m bit word $(l_{m-1},\ldots,l_1,l_0,k_0)$ is extended into W round key words $k_0,k_1,\ldots,k_{W-1}$. From 0 to W-2,, the expansion is done as

l_{i+m-1}<- ((l_i>>> x) addition modulo ki⊕ i

k_{i+1}<- (k_i<<< y) ⊕ l_{i+m-1}

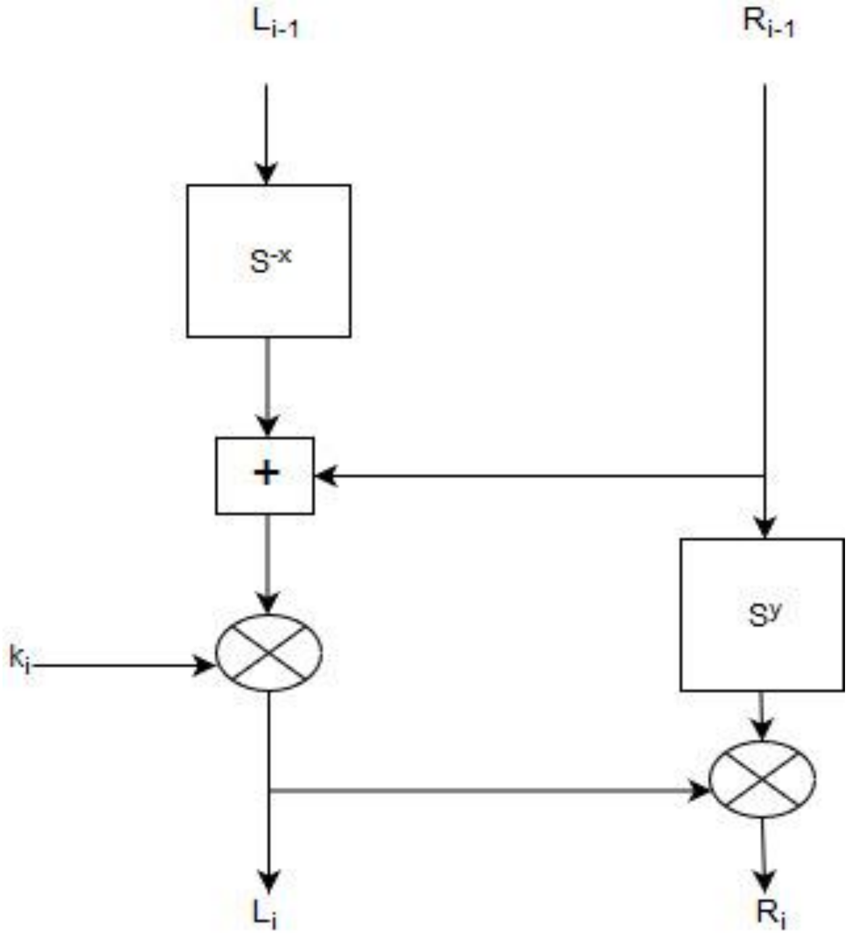


Figure 5.2: SPECK Round Function

The full differential attack on SPECK 2n/mn is implemented by taking the value of n as 32 so block size is 64 and value of m as 4 hence the key size as 128. Therefore, the cryptanalysis is performed on SPECK 64/128.

In differential attack on SPECK, it is assumed that there exists a differential characteristic or differential probability that covers up to r rounds of SPECK having probability let us p. This value of probability p should be greater than 2.2^{-2n} in order to recover the key. The differential cryptanalysis algorithm described in Section 5.3 is used to find out the chosen plaintext P_1 and P_2 and their corresponding cipher text C_1 and C_2 using the encryption algorithm. First a two round attack is executed on SPECK, the input difference (Δa_r, Δb_r) is fixed for the two rounds using the differential characteristic. The actual output values, (a_{r+2}, b_{r+2}) and ($a_{r+2}\oplus\Delta a_{r+2}$, b_{r+2}

$\oplus\Delta b_{r+2}$) are given. Therefore, a guessing process or brute force approach is used to find the value of round key of 2 –rounds k_r and k_{r+1} for decryption. Decryption is carried out in such a way that for the pair (a_{r+2}, b_{r+2}) and ($a_{r+2}\oplus\Delta a_{r+2}$, $b_{r+2}\oplus\Delta b_{r+2}$), the resultant comes out to be (Δa_r, Δb_r). the two rounds of SPECK with the differences in input and output and the actual values of input and output are shown in Figure 5.3

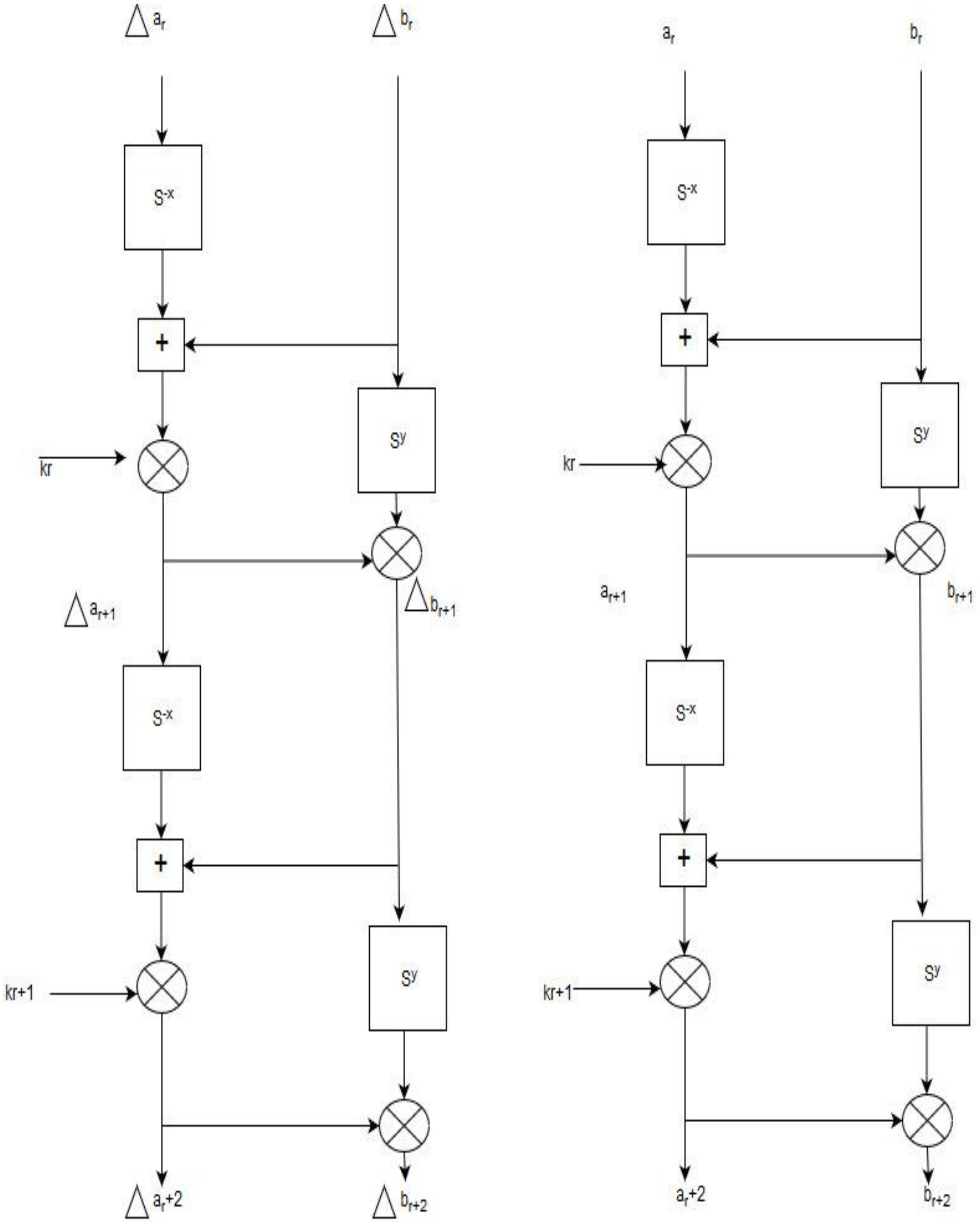


Figure 5.3: Difference in input/output values (Left) and actual input/ output values (Right) in SPECK

5.6 Differential Cryptanalysis of LDS

The input block of n bits is divided into two equal halves. For example, if input text is 64 bit long, it will be divided into 32 bit each represented as L_i and R_i. L_i represents the left sub block and R_i represents the right sub block. The round function of LDS is denoted as

$$F(L_i, R_i) = ((S^{-x} L_i + (S^y (L_i + R_i) \oplus SK_2)) \oplus SK_1, \; S^y (L_i + R_i) \oplus SK_2)$$

Where x and y are the rotation constants. For block size of 64 bit block and key size of 128 bits, the value of x is taken as 7 and the value of y is taken as 3. LDS has 20 operational rounds with each round using two sub key blocks of 32 bit each derived from the original key using the mixing function in generating the round keys.

Applying the same set steps of differential algorithm, differential traits are found in LDS algorithm.

5.7 Comparing Differential Attack Complexity in SPECK and LDS

5.7. 1 Differential Attack Complexity in SPECK

In SPECK, differential characteristic do exist that tend to cover the rounds with $p > 2.2^{-2n}$. The differential attack on SPECK is used to recover the master secret key of mn bit by using brute force technique using r+m rounds. Chosen plaintext required to recover key are $2.p^{-1}$ with the time complexity as shown in equation 14. On the other hand the attack complexity of brute force search is 2^{2n-1}. p.

$$\text{TC} = 2 \times 2^{(m-2)n} \times p^{-1} \qquad (14)$$

The implementation of differential attack on 27-round SPECK 64/128 with the value of m= 4 and n=32, with $p=2^{-60}$. Therefore, $2.p^{-1}$ plaintexts, that is, 2^{61} plaintexts are chosen for gathering the corresponding cipher text pairs. The Time Complexity (TC) of SPECK is calculated using equation 15.

$$\text{TC} = 2 \times 2^{(4-2)32} \times 2^{60} \qquad (15)$$

Therefore, TC= 2^{125}

Moreover, the time complexity measured is faster than the brute force search method on SPECK using 128 bit key by factor of $p.2^{2n-1} = 2^{-60} * 2^{63} = 2^3$. Therefore, it can be concluded that SPECK is vulnerable to differential attack on 19 out of 27 rounds with attack complexity of 2^{125} which is faster from brute force by factor of 2^3.

5.7.2 Differential Attack Complexity in LDS

In LDS, differential characteristic do exist that tend to cover the rounds with $p > 2.2^{-2n}$. The differential attack on LDS is used to recover the master secret key of mn bit by using brute force technique using r+m rounds. Chosen plaintext required to recover key are $2.p^{-1}$ with the time complexity of $2.\ 2^{(m-2)n}.\ p^{-1}$. On the other hand the attack complexity of brute force search is $2^{2n-1}.\ p$.

The implementation of differential attack on 20-round LDS 64/128 with the value of m= 4 and n=32, with $p=2^{-62}$. Therefore, $2.p^{-1}$ plaintexts, that is, 2^{63} plaintexts are chosen for gathering the corresponding cipher text pairs. The TC of LDS is calculated using equation 16

$$\text{TC}= 2 \times 2^{(4-2)32} \times 2^{62} \quad (16)$$

Therefore, TC= 2^{127}

Moreover, the time complexity measured is faster than the brute force search method on LDS using 128 bit key by factor of $p.2^{2n-1}= 2^{-62}*2^{63}= 2^1$. Therefore, it can be concluded that LDS is vulnerable to differential attack on 10 out of 20 rounds with attack complexity of 2^{127}which is faster from brute force by a factor of 2.

After performing differential cryptanalysis on SPECK 64/128 and LDS 64/128, the difference between their time, memory requirements, data, and security level is compared as is shown in Table 5.1

- **Time Complexity:** Here time represents that how much time differential cryptanalysis has taken to break r rounds from the total rounds of operation for a block cipher. If the time requirement of a cipher is more, it is more difficult to break it. Time Complexity of LDS 64/128 is more as compared to SPECK 64/128.

- **Memory requirements:** It includes the required memory to store the plaintext and the corresponding cipher text pairs along with the table required in key guessing phase. Memory requirements of LDS 64/128 are almost same as compared to SPECK 64/128.

- **Data:** It refers to the number of plaintext chosen. For a cipher, if the number of plaintexts required is more to find the differential characteristic, it is less vulnerable to the attack. Data in form of plaintext of LDS 64/128 is more as compared to SPECK 64/128.
- **Attack level:** It is basically for a single key setting the ratio of number of rounds compromised to the total number of operational rounds for a cipher. The security level of SPECK 64/128 and LDS 64/128 are compared and represented in Figure 5.4

Table 5.1: Differential Cryptanalysis Comparison of SPECK and LDS

CIPHER	Time Complexity	Memory	Data	Attack Level
SPECK 64/128	2^{125}	2^{22}	2^{61}	0.70
LDS 64/128	2^{127}	2^{23}	2^{63}	0.50

Result in Table 5.1 shows that proposed LDS takes large time complexity to break, has requirement of more number of plaintexts for its cryptanalysis and offers a high level of security as compared with SPECK.

/

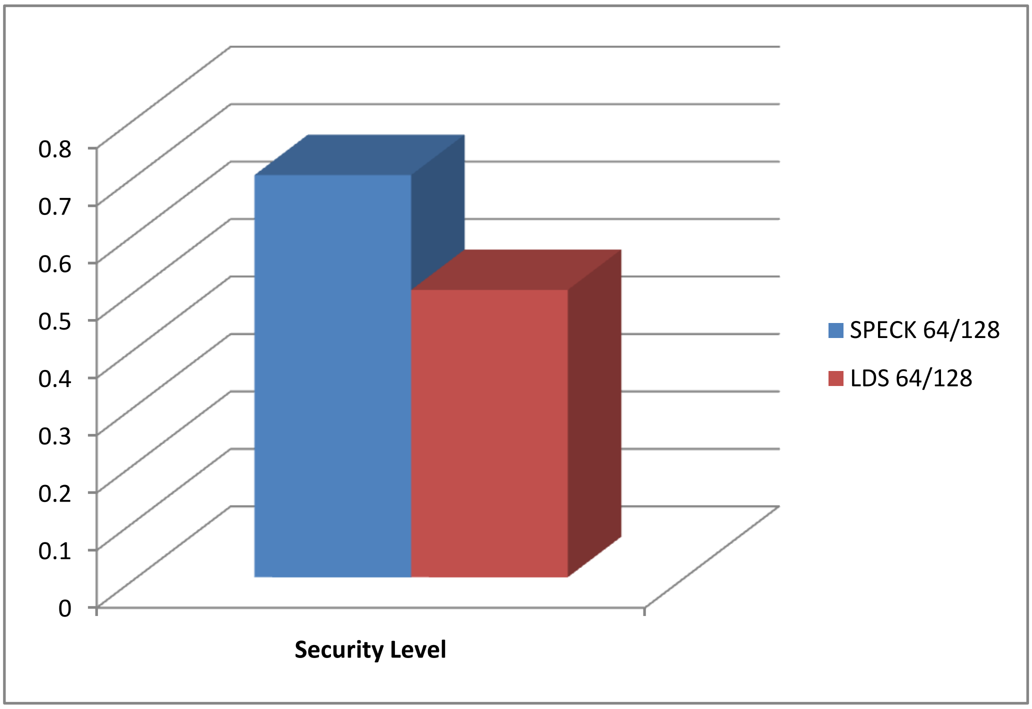


Figure 5.4: Security Level Comparison of SPECK 64/128 and LDS 64/128.

Therefore, LDS 64/128 offers a high security as compared to SPECK 64/128 and is less vulnerable to differential attack.

5.8 Summary

This chapter performs the differential cryptanalysis (Chosen Plaintext Attack) on SPECK 64/128 and LDS 64/128. Differential probability is determined for each variation and differential algorithm is used to implement the attack. Differential algorithm work in two phases, where in first phase, data is gathered from the chosen plaintext. For every plaintext pair the probable cipher texts are determined. In second phase, brute force method is used to find the key of r^{th} round by performing the inverse encryption. Results show that the time required to implement differential attack on LDS 64/128 is more as compared to SPECK 64/128. Memory requirements are almost same. Data in form of plaintext required is more for LDS 64/128 and security level of LDS 64/128 is high as compared to SPECK 64/128 Therefore, it can be concluded that the proposed LDS is less vulnerable to differential attack as compared to SPECK

CHAPTER 6

EFFICIENT DECISION MAKING TO ELEVATE INTEREST OF USER IN IOT USING DATA MINING ON HLSF

The proposed security framework offers authentication and data confidentiality security services. For this it uses a unique mechanism for authentication and a new mechanism is opted as a cryptographic algorithm to provide security. In this chapter, a performance comparison is done among the proposed HLSF and the existing frameworks by executing the whole process of authentication, secure data collection, and finally decision making. Performance is evaluated in context of throughput, latency, and packet delivery ratio.

6.1 Introduction

The security framework designed for an organization must comply with its own standards and policies. A security framework is evaluated by assessing it security features like authentication, confidentiality in this case. The next step for evaluating a framework is to have a check that data collected is authentic or not. The last evaluation measure for a security framework is the synthesis or mining of data to make fruitful decisions. A security framework can be competent if it is able to meet the security requirements and is able to make decisions efficiently in real time.

IoT has lead to revolutionary change in lifestyle of users. Every device that is connected to IoT works in a smart way to make the world technology dependent. IoT is used in numerous applications like inventory, health care, smart homes. Therefore, users expect a great extent of privacy and security from IoT requesting for a security framework. The inbuilt security solutions in IoT are vulnerable to attacks like DoS, spoofing, and many more [130]. The security framework designed for an organization must comply with its own standards and policies. A security framework is evaluated by assessing it security features like authentication, confidentiality in this case. The next step for evaluating a framework is to have a check that data collected is authentic or not. The last evaluation measure for a security framework is the synthesis or mining of data to make fruitful decisions. A security framework can be competent if it is able to meet the security requirements and is able to make decisions efficiently in real time. A security framework tends to provide the overall system security. As shown in Figure 6.1 below , there are 4 layers of operation in IoT.

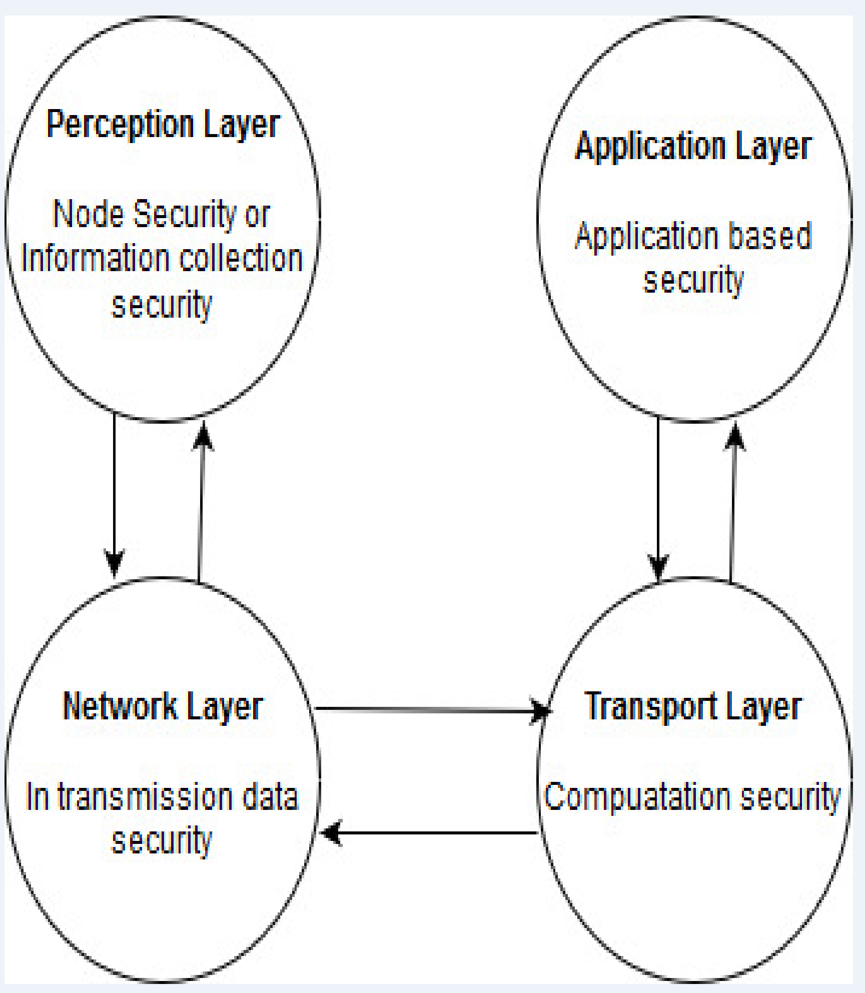

Perception Layer
Node Security or Information collection security
Application Layer
Application based security
Network Layer
In transmission data security
Transport Layer
Compuatation security

Figure 6.1: Security Concerns at Each Layer of IoT

Security frameworks can be designed at every architecture layer of IoT. This work primarily works on the security framework on application layer. Therefore, in next section a detailed analysis is made on existing security frameworks of IoT. Existing frameworks chosen for this study are CoAP and OSCAR. Later, a comparison is done between existing security framework CoAP, OSCAR and the proposed security framework HLSF. Finally, the association based mining approach is applied on CoAP and HLSF to find the framework having an efficient way of decision making by evaluating its precision, recall, and accuracy.

6.2 Existing Security Frameworks in IoT

After conducting the theoretical and analytical study, this section describes the current state of existing security frameworks in IoT. There are number of existing security frameworks in IoT, may be, for achieving the same target and following the different approaches. Each framework is designed based on the same level of expectancy from them as specified below

- Software reliance in each framework to carry out the whole process
- Set of protocols required to initiate and setup the communication among the devices.
- Contribution of the security framework in maintaining the security and privacy in IoT.

Considering the expectancy from a security framework and finding the authentication, cryptographic solutions used by the framework is the prime concern of this section. The frameworks analyzed in this study are Constrained Application Protocol (CoAP) framework, Object Security Framework (OSCAR) for IoT.

6.2.1 Constrained Application Protocol (CoAP) Framework

CoAP was proposed by constrained restful environment working group (CORE) in IETF. CoAP works for constrained devices at the application layer. For communication, IPV6 over Low Power Personal Area Network (6LoWPAN) provides the usage of IPv6 communication among the sensing devices. IoT devices can communicate in CoAP by using User Datagram Protcol (UDP) at transport layer and 6LoWPAN [131].

6.2.1.1 Architecture

Interactions in constrained network like IoT running CoAP can be either between devices or client/server where one of the devices can act as client and one dedicated device act as a server [132]. CoAP works in similar manner as Hyper Text Transfer Protocol (HTTP) in internet. CoAP makes use of Uniform Resource Location (URL) such as “coap://ipv6host:port/abc. CoAP implementation can also allow a CoAP server to act as a proxy gateway to access resources available on CoAP server using HTTP client. Figure 6.2 shows that integration of CoAP with the internet.

CoAP itself is the internal network that means a CoAP client request can only be processed by the CoAP server. CoAP, otherwise, can be extended and can process HTTP client request using CoAP/HTTP mapping process as CoAP acts as a subset of HTTP. 6LoWPAN border router (6LBR) can be used to establish this connection as shown in Figure 6.2

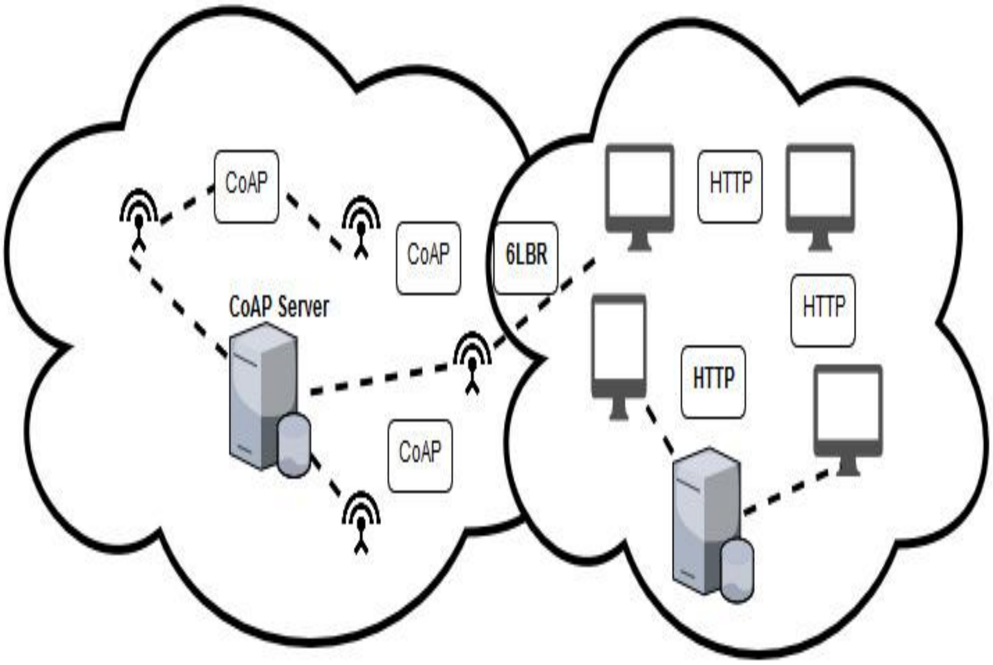


Figure 6.2: CoAP Network

At the application layer CoAP framework uses CoAP protocol, at transport layer CoAP uses UDP, and at the network layer 6LoWPAN is used [133]. At network layer 6LoWPAN is used as it commercializes with the constrained networks like IoT. At transport layer UDP is used for fast transmission as compared to the reliable counter measure Transmission Control Protocol (TCP). This is due to the reliability mechanism offered by the message layer of CoAP.

At application layer, CoAP operates at distinguished layers. The application layer deploying CoAP is categorized into two sub layers referred as Request/Response and Message Layer as shown in Figure 6.3 below.

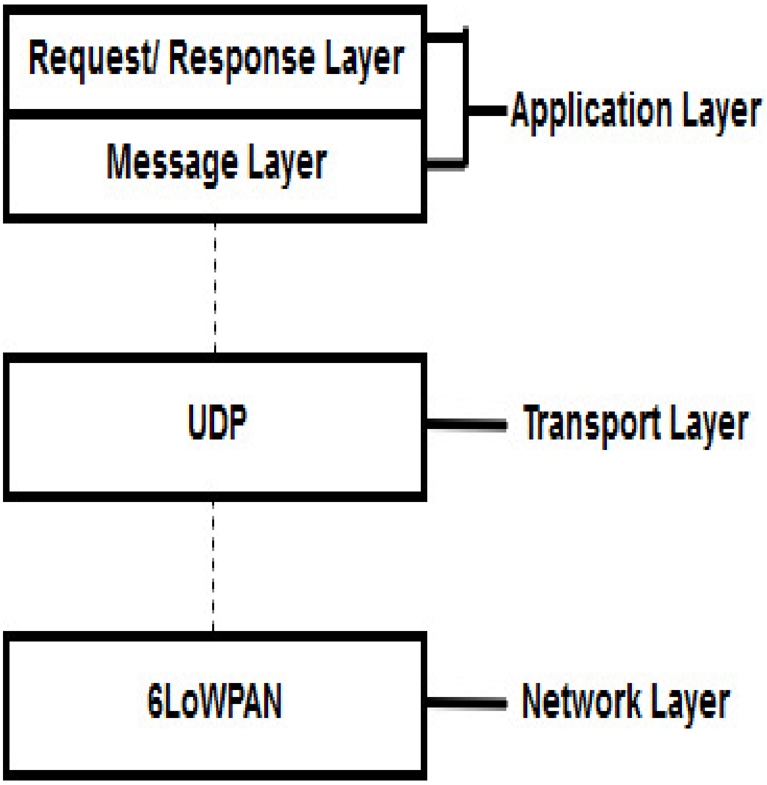


Figure 6.3: CoAP Protocol Stack

The responsibility of the request/response layer is to employ methods like GET, PUT, POST, DELETE for accessing the resources in CoAP network [134-136]. The number of requests and the mapping among their semantically correct responses also forms the responsibility of this layer. The ultimate task of message layer lies in supporting reliability with use of three types of messages as follows

- **Confirmable Message:** Here acknowledgement is required as response
- **Non Confirmable Message:** No acknowledgment is required as the response.
- **Reset:** If the message that is confirmable is received and cannot be further. Processed.

6.2.1.2 Security Features

CoAP offers security features by using Datagram Transport Layer Security (DTLS) over UDP instead of TCP. DTLS is designed to offer end to end security. As it runs with UDP, it can be used in numerous constrained applications like Voice over IP (VoIP), real time communication. DTLS in CoAP offers security features such as authentication, confidentiality, integrity, key sharing mechanism [137]. This research work emphasize on the authentication and the confidentiality security aspect of CoAP. CoAP comes in four security options, namely No Sec, Pre Shared Key, Certificates and Raw Public key. For all the four security options authentication and cryptography solutions are analyzed.

- **No Sec Option:** The first option No Sec does not use DTLS for security, hence does not support any security feature. Here the packet transmission is done as normal UDP datagram as shown in Figure 6.4.

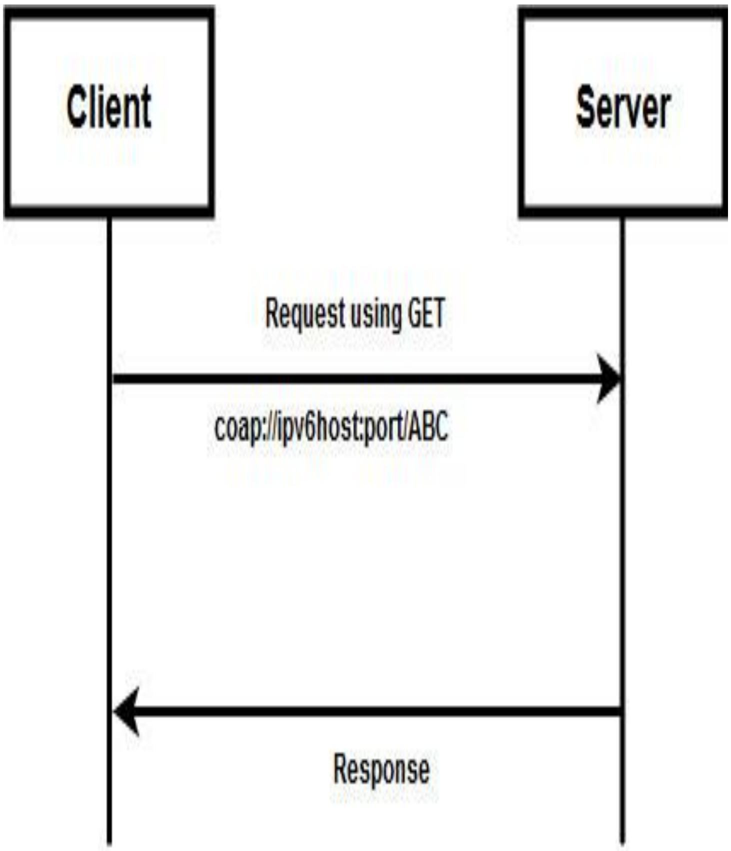


Figure 6.4: No Sec Option of CoAP

- **Pre Shared Key Option:** In Pre Shared key option a secret key is shared between the devices that are part of IoT. Simple Diffie-Helmen (DH) algorithm can be used for key exchange among the devices. For authentication shared secret key can be used. Client sends a message that is encrypted using secret key that is only opened by same secret key at receiving side as represented in Figure 6.5. Cryptographic security solution to offer confidentiality is provided by using AES in Counter Cipher Block Chain Mode (CCM). Each packet is encrypted using the same shared secret key.

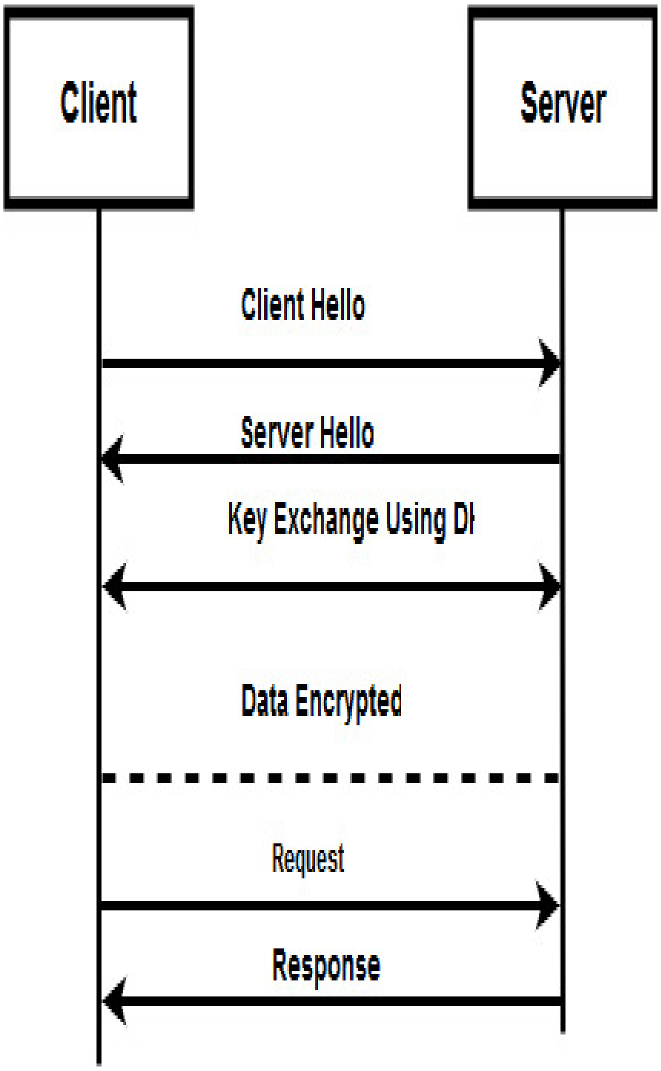


Figure 6.5: Pre Shared Key Option of CoAP

- **Certificate Option:** In this option, Certificate Authority (CA) is used to validate the certificates owned by the client devices or the server. For authentication among devices acting as clients or server RSA algorithm is used as shown in Figure 6.6. Cryptographic security solution to offer confidentiality is provided by using AES in Cipher Block Chain Mode (CBC).

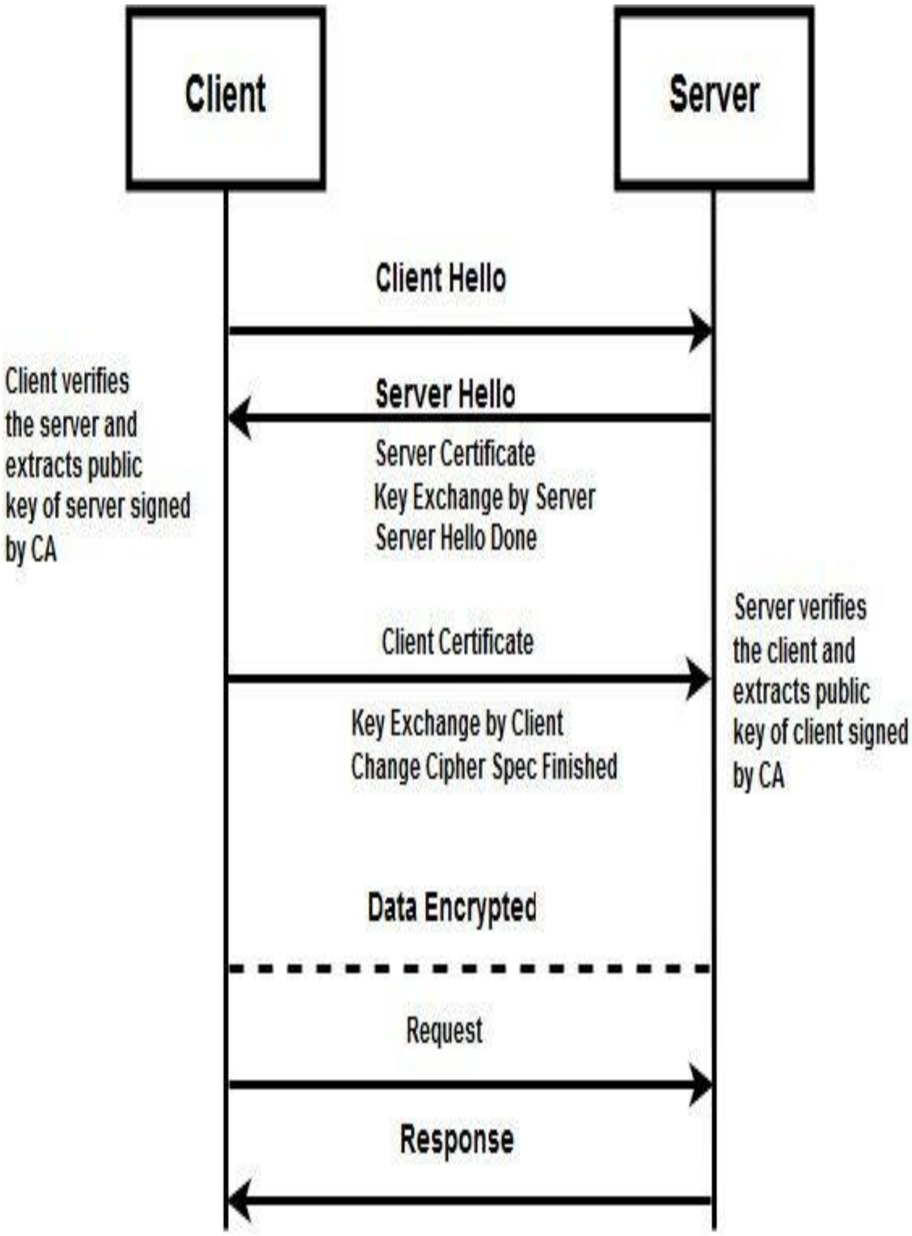

Client
Server
Client Hello
Client verifies the server and extracts public key of server signed by CA
Server Hello
Server Certificate
Key Exchange by Server
Server Hello Done
Client Certificate
Server verifies the client and extracts public key of client signed by CA
Key Exchange by Client
Change Cipher Spec Finished
Data Encrypted
Request
Response

Figure 6.6: Certificate Option of CoAP

- **Raw Public Key Option:** Devices in this option make use of the public keys of other devices in the network, but there is no central authority available to validate the public key as shown in Figure 6.7. For authentication, elliptic curve with diffie-helmen and ephemeral key (ECDHE) is used. Cryptographic security solution to offer confidentiality is provided by using AES in CCM mode is used.

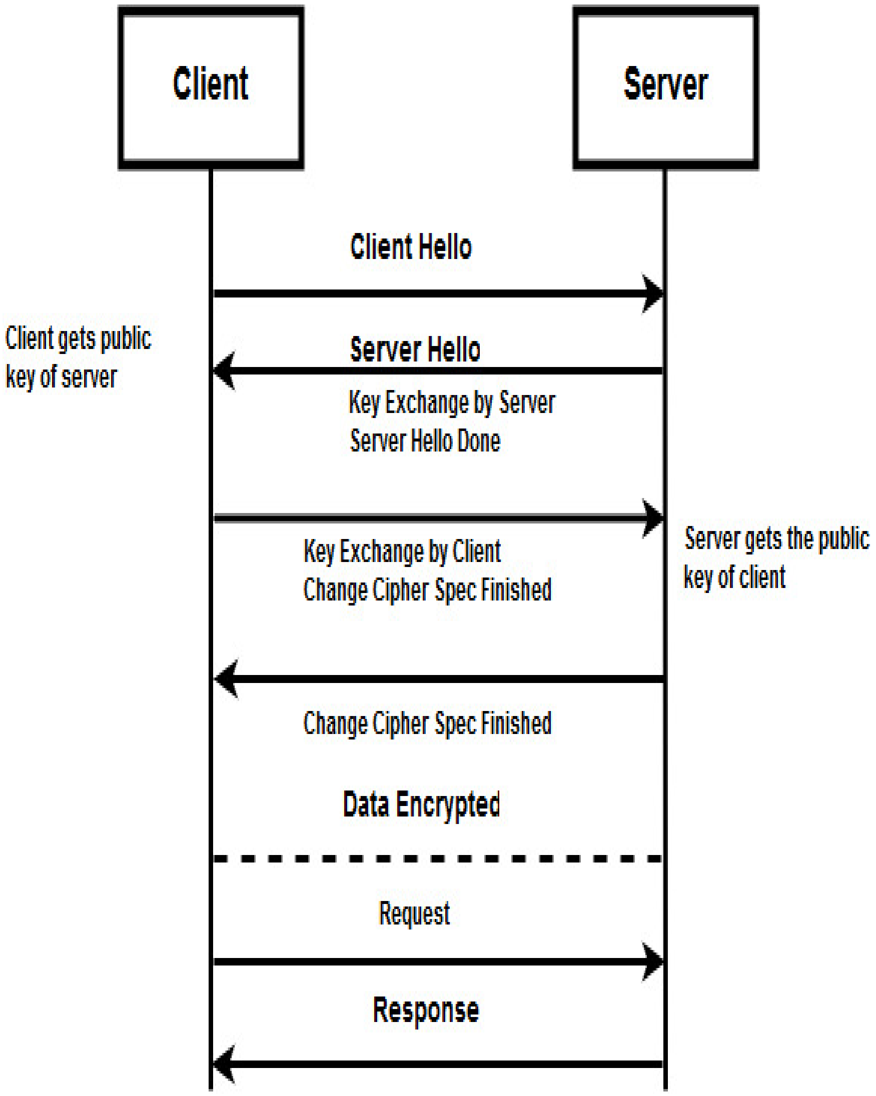


Figure 6.7: Raw Public Key Option of CoAP

6.2.2 Object Security Framework for IoT (OSCAR)

OSCAR framework works on consumer-producer model. In IoT, consumers are the end devices such as human accessories used by users, actuating devices that intake the data from the producers such as sensors, smart watch, smart meters, and motion sensors. The responsibility of offering security in OSCAR lies with the producers. Security is provided to the data at rest or during transmission. The major emphasis of OSCAR is on the object/device security. Global applications like smart city work on OSCAR as there are number of consumers/clients requesting the services of constrained servers [138]. Figure 6.8 represents the OSCAR framework as the typical consumer-producer model.

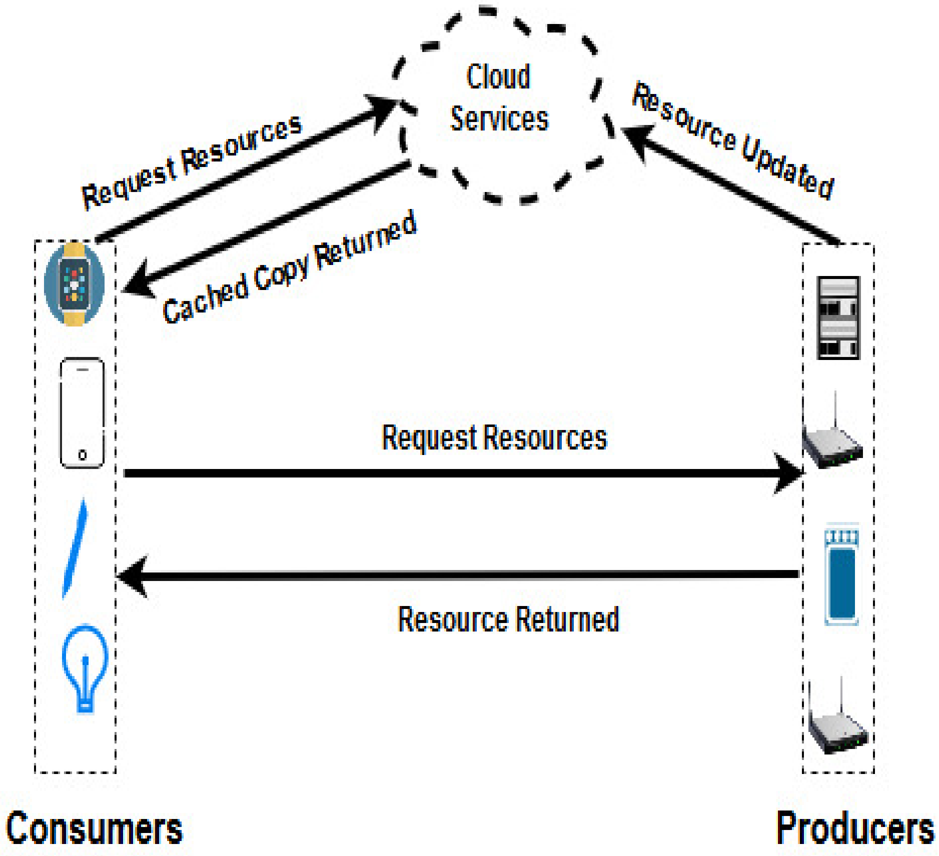


Figure 6.8: OSCAR Framework

OSCAR needs authorization servers to restrict the access of resources by the consumers. For authentication OSCAR uses simple concept of elliptic curve digital signatures. Cryptographic security solution to offer confidentiality is provided by using AES in CCM mode is used.

6.3 Performance Comparison of CoAP, OSCAR, and HLSF

This section evaluates and compares the performance of existing frameworks CoAP, OSCAR with the proposed framework HLSF. First, the security effectiveness of the frameworks including authentication, data collection, data security is tested by finding the memory requirements, energy overhead, computational overhead, throughput, packet delivery ratio and latency. Later, the effectiveness of overall framework using data mining and decision making is tested in context of accuracy, precision and recall. Considering the heterogeneous nature of frameworks, certain research assumptions are made and realized for the performance evaluation.

6.3.1 Memory Requirements

The Read Only Memory (ROM) and Random Access Memory (RAM) is evaluated for CoAP, OSCAR, and proposed HLSF using COOJA as the simulator. Figure 6.9 represents the memory requirements in percentage by considering the total available memory in CoAP, OSCAR, and HLSF.

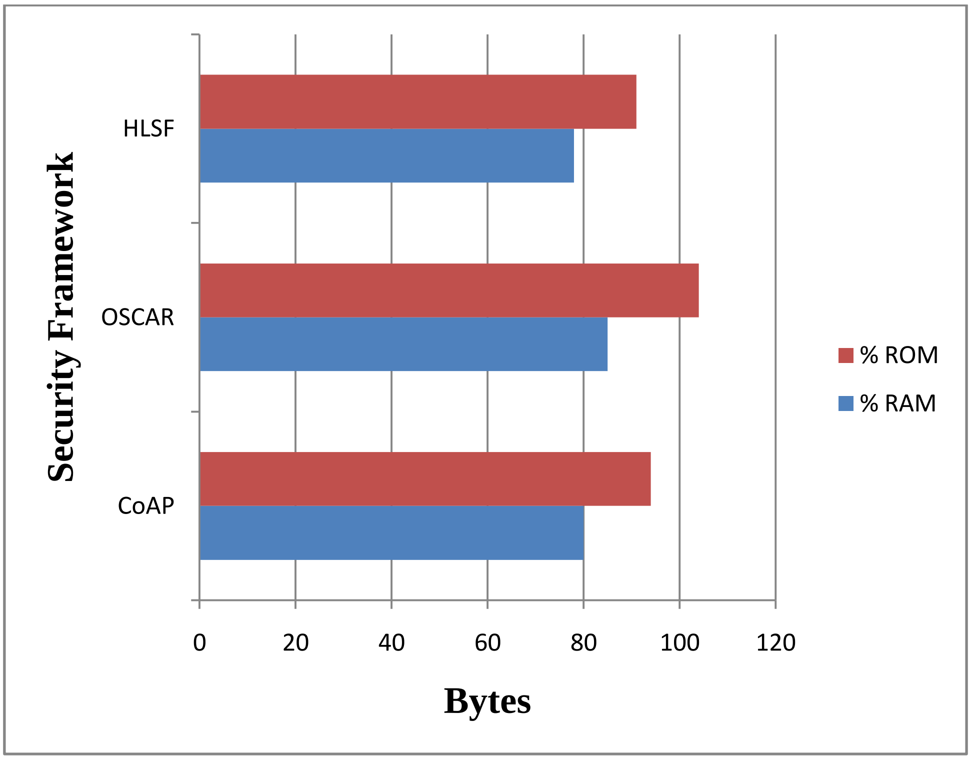


Figure 6.9: Percentage Memory Requirement for CoAP, OSCAR, and HLSF

HLSF has RAM requirements 18% less than OSCAR and 5% less than CoAP. ROM requirements of HLSF are 25% less than OSCAR and 5% less than CoAP. Therefore, Figure 6.9 above concludes that the requirements for memory either RAM or ROM for proposed HLSF is less in comparison to CoAP and OSCAR.

6.3.2 Energy Overhead

The energy overhead of security framework has a direct impact on the lifetime of the sensors and will ultimately impact the transmission rate of the application. Therefore, if the energy overhead increases for a particular application, its lifetime gradually decreases. A more complex security framework possesses more energy overhead. Figure 6.10 represents the energy overhead calculated in mili joules (mJ) for CoAP, OSCAR, and HLSF.

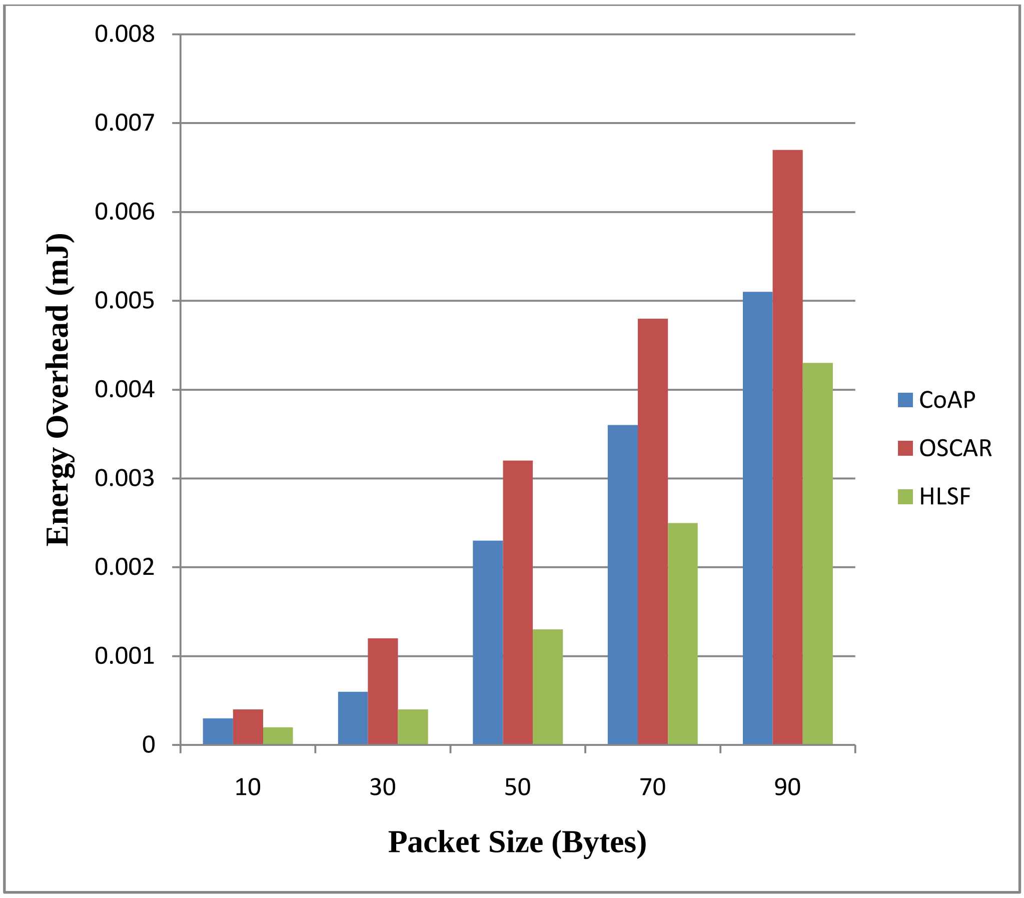


Figure 6.10: Energy Overhead for CoAP, OSCAR, and HLSF

HLSF has 35% to 67% less energy overhead than OSCAR considering the varying packet size. HLSF has 15% to 43% less energy overhead than CoAP considering the varying packet size Therefore, Figure 6.10 above concludes that the energy overhead for proposed HLSF is less in comparison to CoAP and OSCAR.

6.3.3 Computational Overhead

Computational overhead is the excess time required by a security framework for offering security used in IoT application. Figure 6.11 represents the computational overhead evaluated in milli second (ms) for CoAP, OSCAR, and HLSF.

HLSF has 18% to 36% less computational overhead than OSCAR considering the varying packet size. HLSF has 7% to 21% less computational overhead than CoAP considering the varying packet size Therefore, Figure 6.11 below concludes that the computational overhead for proposed HLSF is less in comparison to CoAP and OSCAR.

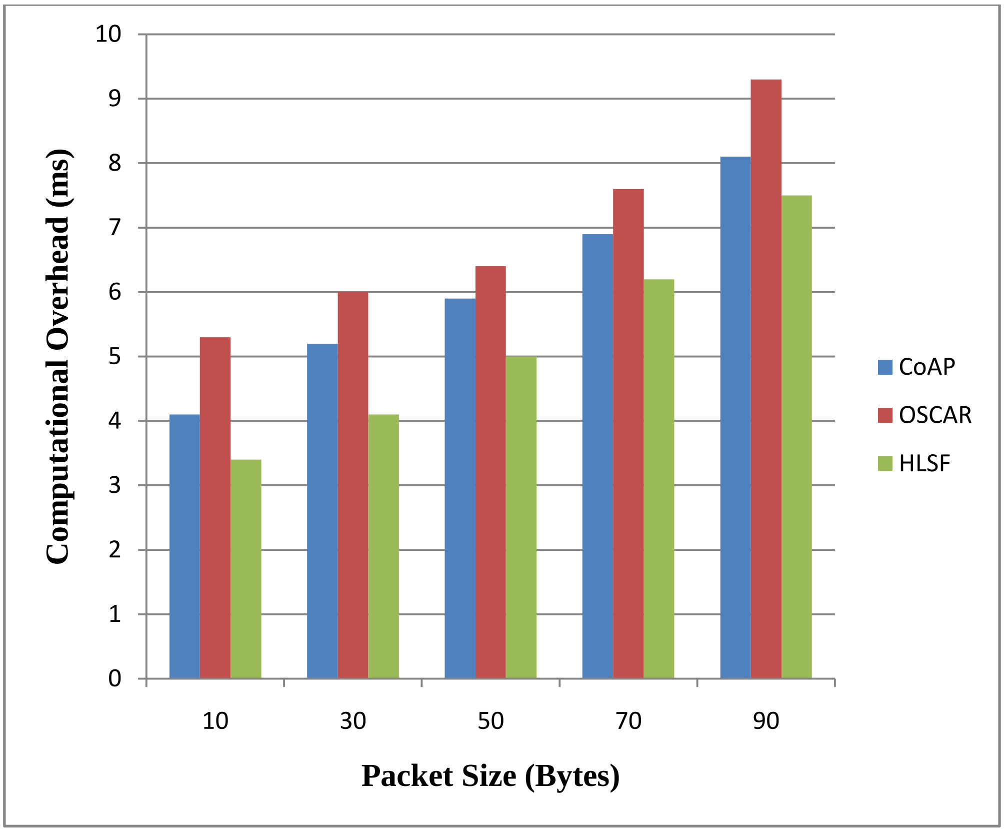


Figure 6.11: Computational Overhead for CoAP, OSCAR, and HLSF

6.3.4 Throughput

Throughput is the number of packets sent over in one second. Throughput gets impacted by the type of security framework used for an application scenario. An application do not offering any security will tend to have more throughput than a security oriented application. Figure 6.12 represents the throughput of CoAP, OSCAR, and HLSF.

HLSF has 11% more throughput than OSCAR and 2% more throughput than CoAP. Therefore, Figure 6.12 below concludes that the throughput for proposed HLSF is high in comparison to CoAP and OSCAR. It means that HLSF can send more bits in one second after offering security solutions like authentication, confidentiality and integrity.

Throughput cannot be considered as the prime criteria for determining the effectiveness of a framework as throughput may be more for a unsecure framework too. To effectively evaluate a framework its packet delivery ratio should be evaluated that gives the actual packets received at destination.

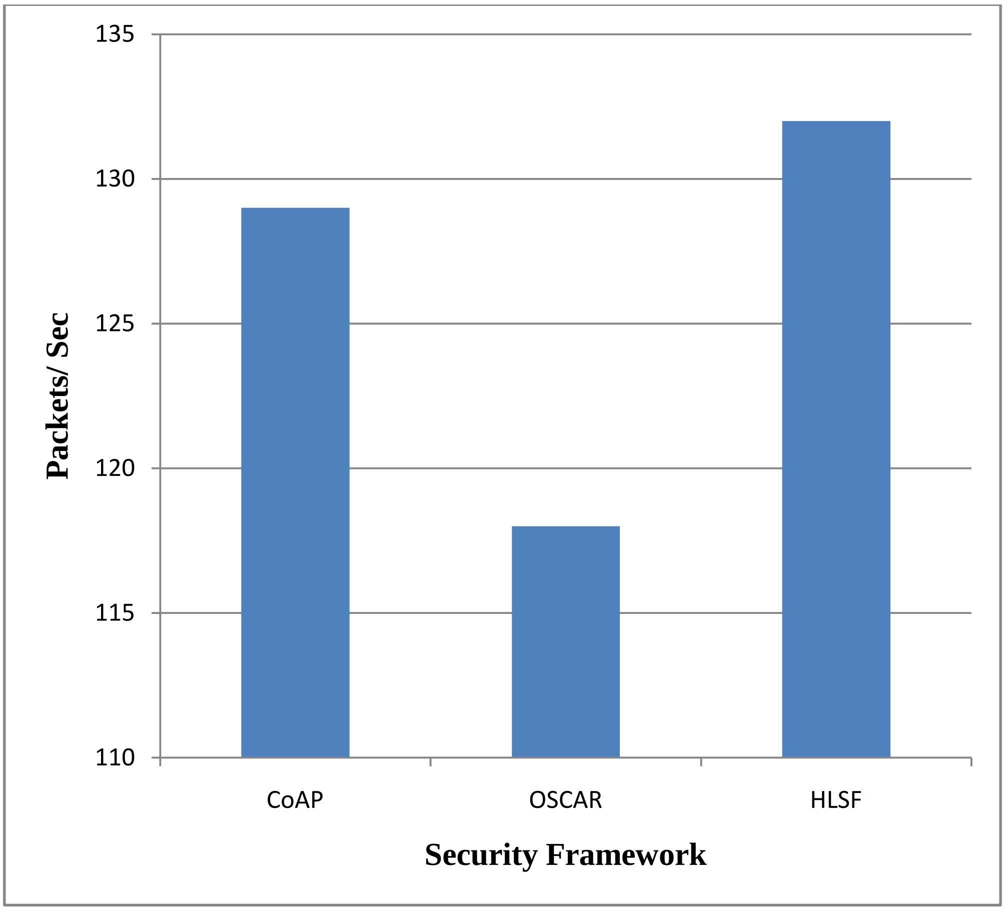


Figure 6.12: Throughput for CoAP, OSCAR, and HLSF

6.3.5 Packet Delivery Ratio

To measure the effectiveness of communication, the number of packets received intact at the destination to the total generated packets at source is referred as packet delivery ratio. If security mechanism is strong the packet delivery ratio is going to be more because it may happen a intruder in between will acquire the packets and do not let them reach the destination. Figure 6.13 represents the packet delivery ratio of CoAP, OSCAR, and HLSF.

The packet delivery ratio is calculated in percentage that is how many packets are received at destination location to how many packets that are generated at the sending location. HLSF has 16% more packet delivery ratio than OSCAR and 10% more packet delivery ratio than CoAP Therefore, Figure 6.13 below concludes that the packet delivery ratio for proposed HLSF is high in comparison to CoAP and OSCAR. That means more number of packets reach the destination securely and in correct form.

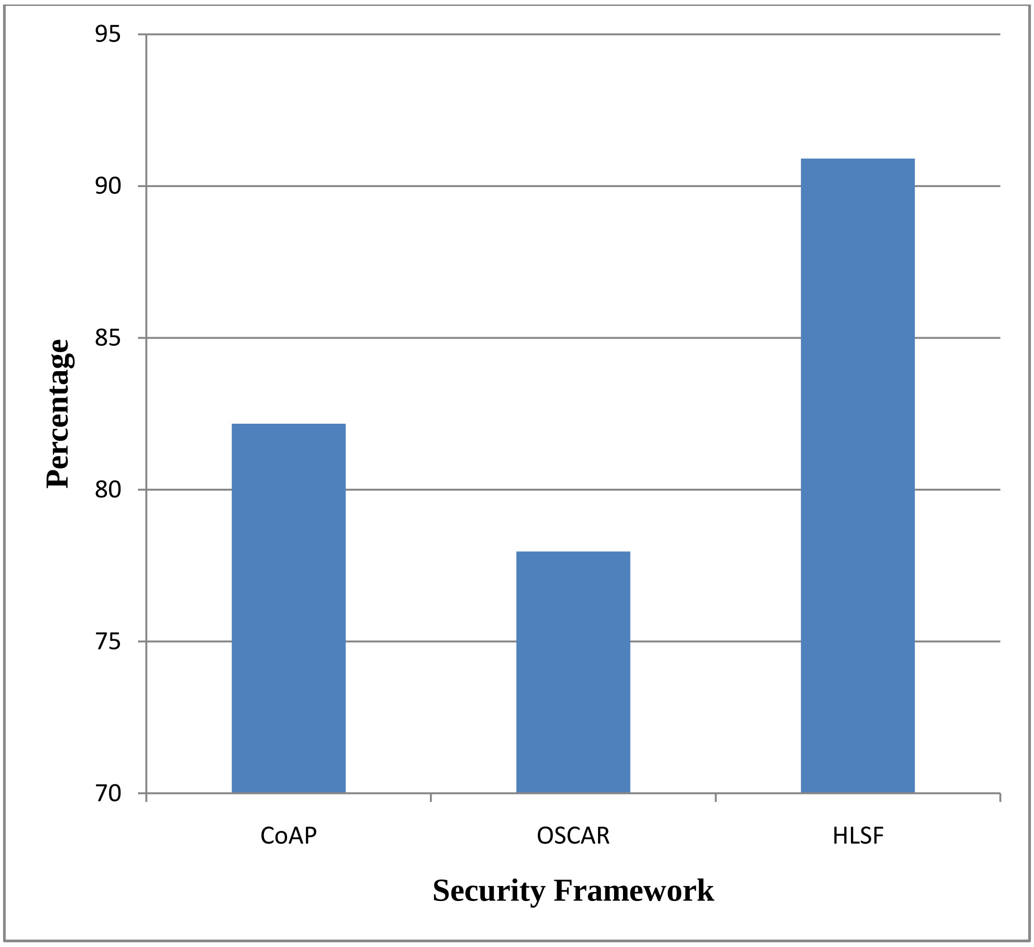


Figure 6.13: Packet Delivery Ratio for CoAP, OSCAR, and HLSF

6.3.6 Latency

The time taken to send a packet from a specific source to destination is denoted as latency. Latency is calculated in milliseconds (ms). To be an effective framework the latency that is the overall time taken for a framework should be less. Figure 6.14 represents the latency of CoAP, OSCAR, and HLSF.

The latency is calculated by finding the number of overall milliseconds taken by an framework to send a packet from one sending location to the destination location. HLSF has 75% less latency than OSCAR and 50 % less latency than CoAP. Therefore, Figure 6.14 below concludes that the latency for proposed HLSF is less in comparison to CoAP and OSCAR. That means that amount of time taken to offer authentication services, confidentiality services and integrity services is less in HLSF as compared to other two frameworks. If HLSF is having less execution time draws that it is less complex and hence is lightweight in nature.

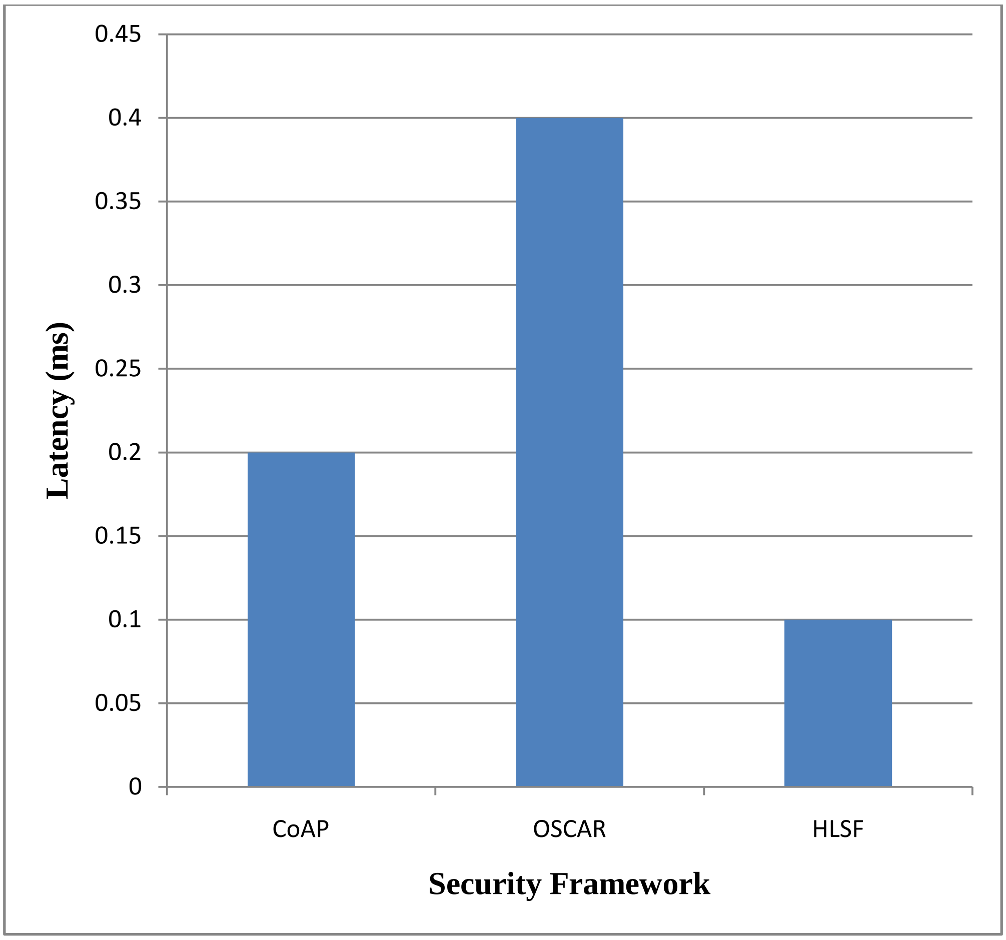


Figure 6.14: Latency for CoAP, OSCAR, and HLSF

6.4 Performance Evaluation of Data Mining Algorithms

As described in Section 2.5, multiple techniques exist for applying data mining on the data collected in an application scenario. In literature, classification, clustering and association rule based mining techniques are studied.

From the study, it was devised that different algorithms working on the mentioned techniques can applied in IoT. For classification kNN algorithm is implemented [139], for clustering k-means [140] and for association rule based mining FP (frequent pattern) growth algorithm is used [141]. The three methods of mining are compared in this study to figure out the best operating and fruitful decision maker in IoT. The performance comparison among kNN, k-means and FP algorithms is made on the basis of precision, accuracy and the recall value.

The one performing well will be used in HLSF and CoAP in order to evaluate their performance.

6.4.1 Precision

Precision is the ratio of the appropriate occurrences collected to the total number of collected occurrences. If the number of collected appropriate occurrences is more, the precision of a DM technique is more. The precision of kNN algorithm, k-means, and FP algorithm is evaluated as shown in Table 6.1 and is represented through Figure 6.15 below.

Table 6.1: Precision Based Analysis of Knn, K-Means and FP In IoT

Iterations	kNN	k-means	FP
1	93.1613	94.5843	94.9503
2	95.1015	96.5245	95.8905
3	94.9231	96.3461	97.7121
4	93.7143	96.1373	96.5033
5	95.4127	96.836	97.202
6	93.082	95.505	94.871
7	96	96.923	97.289
8	96	96.923	97
9	92.9153	93.8383	97
10	91.75	92.673	94.87

With the varying number of iterations, the percentage of precision of FP algorithm is more as compared to kNN and k-means cluster algorithm as concluded in Figure 6.15.

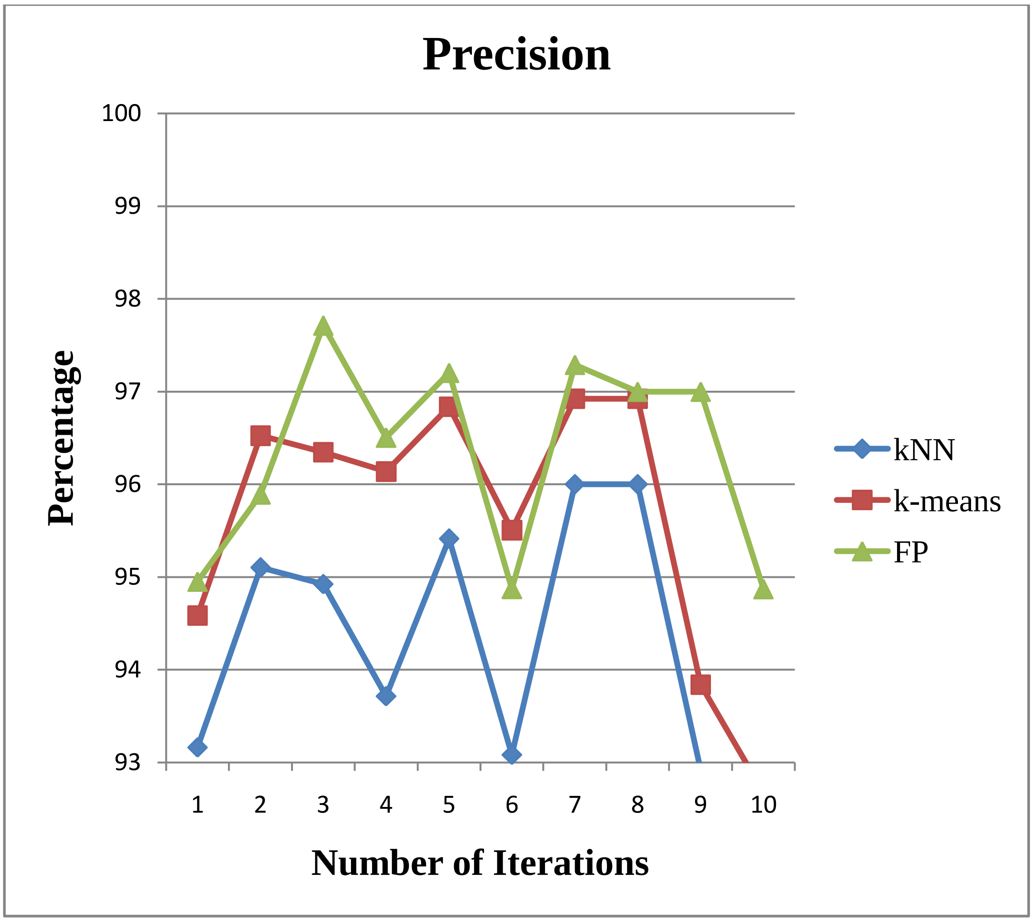


Figure 6.15: Performance Comparison of kNN, k-means and FP on the basis of Precision

6.4.2 Accuracy

Accuracy is the number of collected appropriate occurrences without any systematic errors like Type 1 and Type 2 errors. If the number of collected appropriate occurrences without any error is more, the accuracy of a DM technique is more. The accuracy of kNN algorithm, k-means, and FP algorithm is evaluated as shown in Table 6.2 and is represented through Figure 6.16 below.

Table 6.2: Accuracy Based Analysis of kNN, K-Means and FP in IoT

Iterations	kNN	k-means	FP
1	94	97	98
2	95	98	98

3	96	97	98
4	94	96	97
5	97	97	98
6	95	97	98
7	98	97	98
8	98	97.43	98
9	94	98	97.67
10	92	93	95

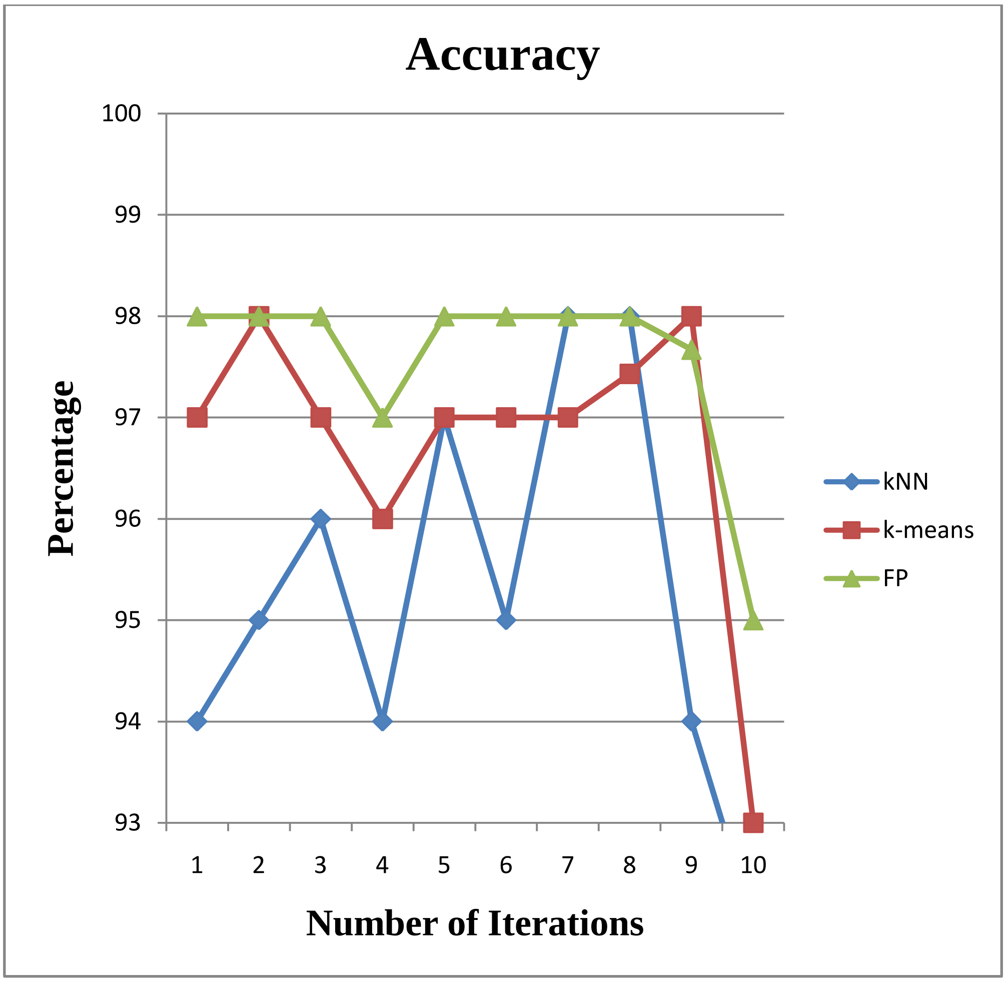


Figure 6.16: Performance Comparison of kNN, k-means and FP on the basis of Accuracy

With the varying number of iterations, the percentage of accuracy of FP algorithm is more as compared to kNN and k-means cluster algorithm.

6.4.3 Recall

Recall is the ratio of the appropriate occurrences collected to the number of appropriate occurrences that would have been collected. If maximum number of appropriate occurrences is collected, the recall of a DM technique is more. The recall of kNN algorithm, k-means, and FP algorithm is evaluated as shown in Table 6.3 and is represented through Figure 6.17 below.

Table 6.3: Recall Based Analysis of kNN, k-means and FP in IoT

Iterations	kNN	k-means	FP
1	95.3333	96.33333	96
2	94.5294	95.52941	96
3	95	96	97
4	96.5294	97.5241	98
5	96.762	96.671	97.234
6	96	97	98
7	95	96	96.3871
8	95	95.456	96
9	96.7724	97.7724	98
10	96	97.3419	97.5681

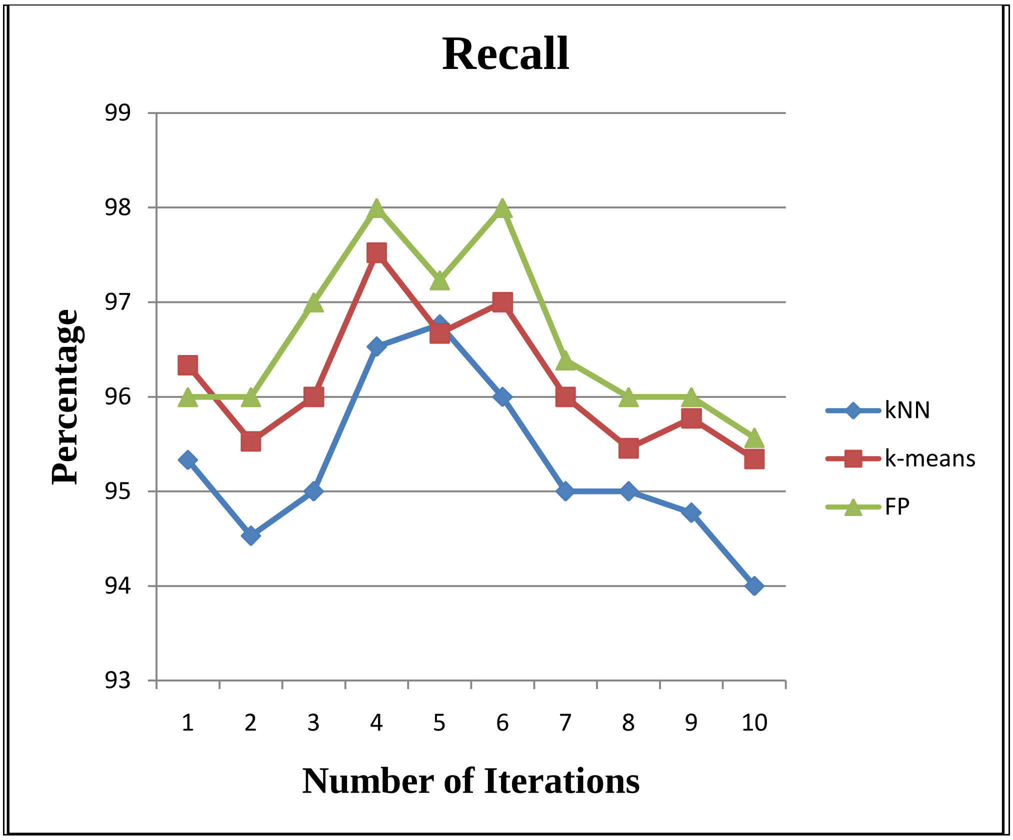


Figure 6.17:Performance Comparison of kNN, k-means and FP on the basis of Recall

With the varying number of iterations, the percentage of recall of FP algorithm is more as compared to kNN and k-means cluster algorithm. From the evaluated performance parameters it can be concluded that the precision, accuracy and recall of FP algorithm is more as compared to other algorithms

6.5 Applying Data Mining on Data Collected using HLSF

FP algorithm is applied on data collected using HLSF, and then the precision, accuracy and recall are evaluated as represented through Figure 6.18. The resultant values over multiple iterations are shown in Table 6.4 below.

Table 6.4: Precision, Accuracy, and Recall Based Analysis of HLSF

Iterations	Precision	Accuracy	Recall
1	99.147	99.437	99.5

2	99	99.21	99.3
3	98.345	99	99
4	98.763	98.812	98.9
5	98	98.3	98.4
6	98	98	98
7	97.824	98	97.6
8	97	97.87	97.3
9	97	97.7	97
10	96.967	97	96.6

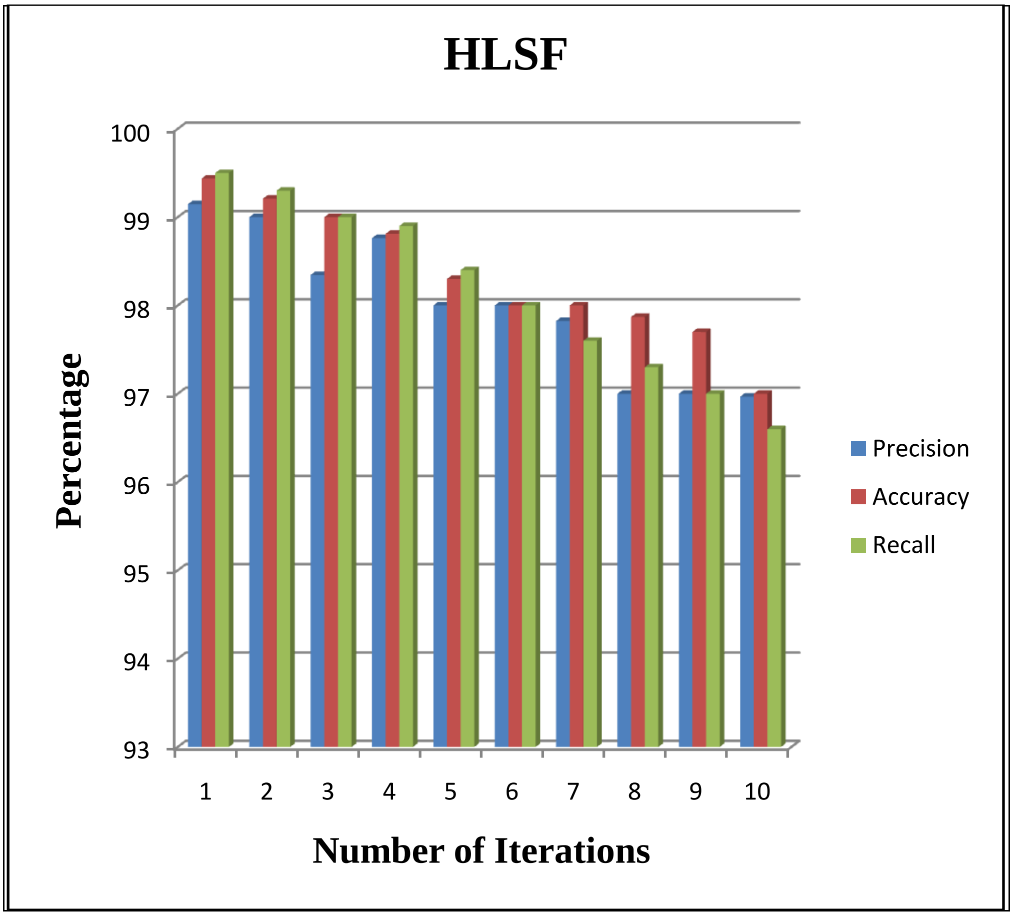


Figure 6.18: Precision, Accuracy, Recall of HLSF

In this research work, IoT scenario is implemented in two ways. One, when data is collected using security mechanism as HLSF. Second, when data is collected using

CoAP as security mechanism. Later, data mining is applied in both the scenarios to find out the precision, accuracy, and recall. Figure 6.19 represents the precision percentage comparison, Figure 6.20 represents the accuracy percentage comparison, and Figure 6.21 represents the recall percentage comparison in two scenarios with HLSF security and CoAP security.

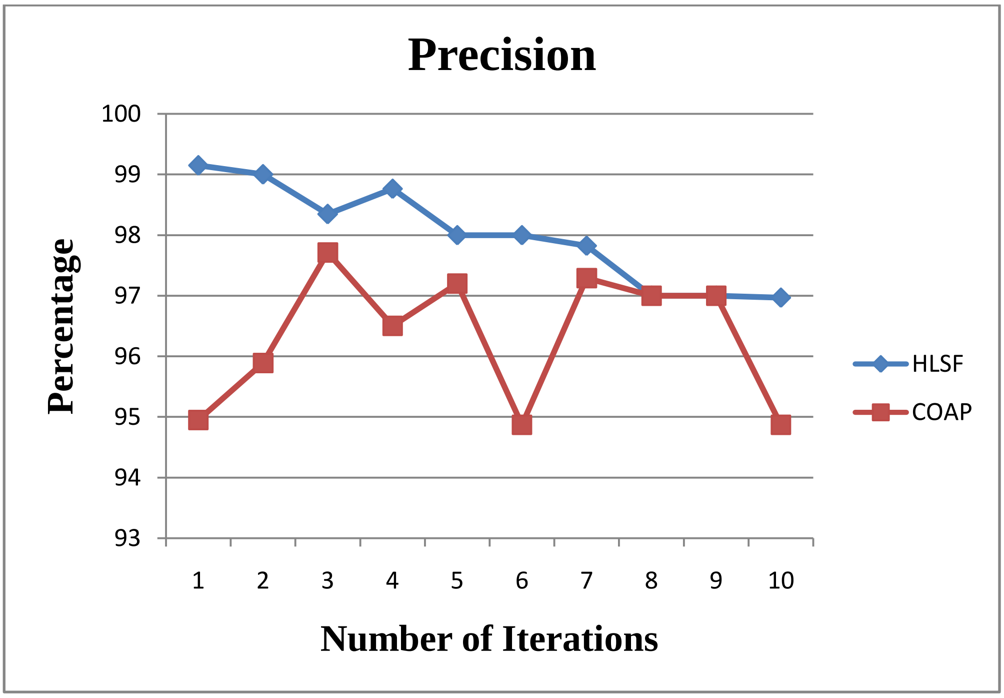


Figure 6.19: Precision Comparison of HLSF with CoAP

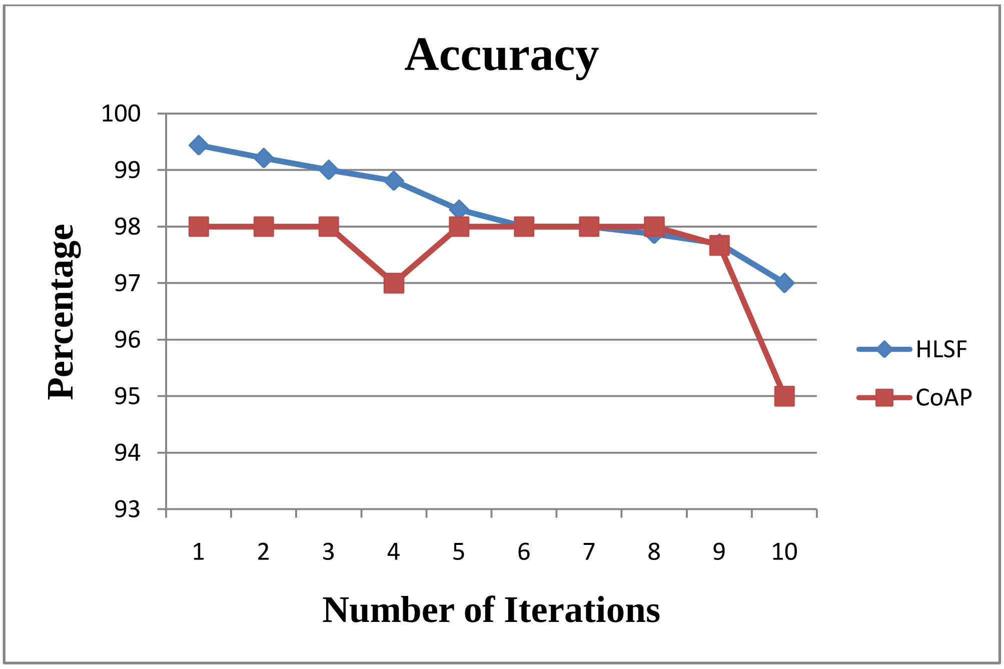


Figure 6.20: Accuracy Comparison of HLSF with CoAP

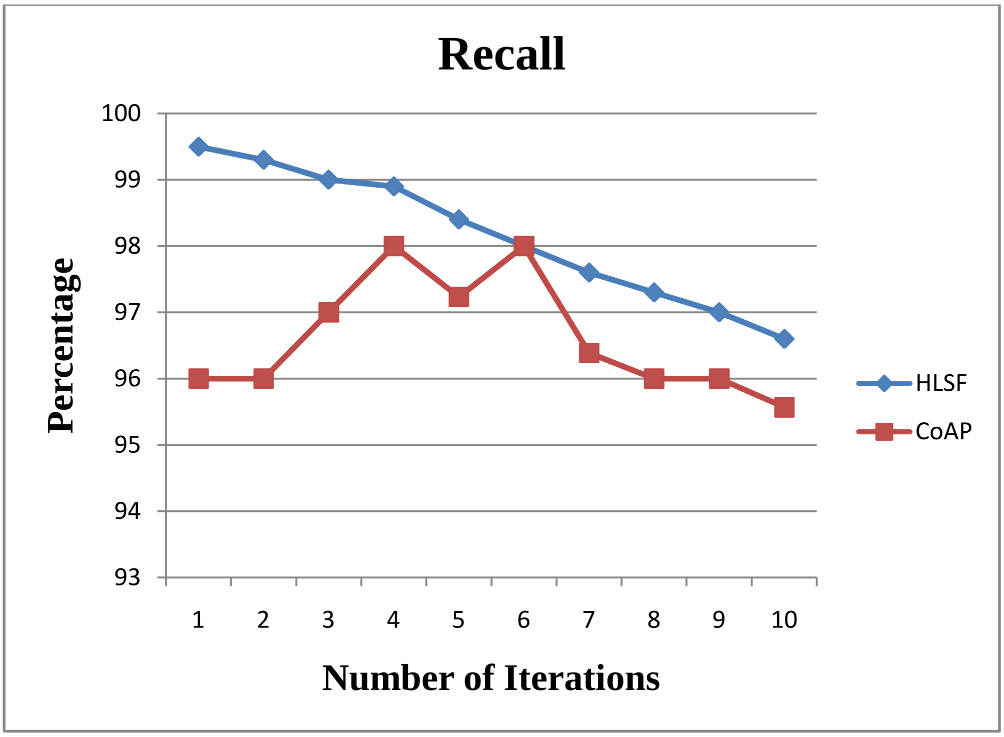


Figure 6.21: Recall Comparison of HLSF with CoAP

6.6 Summary

The results and discussions made in this chapter are on applying data mining techniques in IoT scenario using HLSF framework. Initially, the existing data mining techniques kNN, k-means and FP are compared by analyzing them on the basis of Precision, Accuracy and Recall. Results show that FP technique outperforms over other mining techniques. Later, FP is applied on first: IoT scenario using HLSF framework and later on IoT scenario with CoAP as security mechanisms. Results prove that proposed framework HLSF is better in generating data on basis of precision, accuracy and recall and hence lead to effective decision making.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

Internet of things (IoT) has now become a fascinating system that improves information technology for its use in homes, cities and medical sectors. IoT works as an extension of internet to realize interconnections among every day object based on platform independent communication protocols. Object forming IoT must possess sensing, communication, and computation capabilities leading to a convenient as well as economical assistance for society. Interaction among heterogeneous objects enhances the security vulnerabilities in IoT. With security as a prime concern, communication in IoT need to maintained using a secure mechanism to protect the system from attacks.

7.1 Conclusion

In last few years, with gradual growth in Information and communication technology (ICT), the epoch of smartly operated things has evolved at exponential rate. Thus, IoT is taking its place in homes, office, cities, agriculture, business and many more at a very fast pace. One of the promising technologies using the things smartly is IoT. These smart things are capable of communication as well as can make decision depending on persuaded processing. IoT involving things for communication may belong to users of different functionalities, which raises a concern on privacy and security of data from the diverse users. Therefore, one of the significant contributions in the advancement of IoT is security.

IoT has evolved as an innovation of next generation in this world of smart devices. IoT intends to provide services for data collection, data management, data and device security for application development. IoT supports interconnection of various heterogeneous things such as sensors, cameras, devices, smart phones for offering automation in home, health care, industry, or military. Things or devices in IoT communicate and compute to make our lives comfortable and safe. In inventory automation, real time check on items, their information management, status management, monitoring can be carried out using IoT. The huge amount of data that flows among the devices in the network demands for a security framework that ensures authentication, authorization, integrity and confidentiality of data. IoT has evolved as the fast adapted technology among the users due to its anywhere anything connectivity. As anything can connect, this may impose a serious impact of security of data for these connected devices.

Smart things in IoT are enabled to compute, communicate, and take decisions related to any network activity. Therefore, this requires a secure solution of communication between heterogeneous devices.

Number of security solutions is available in literature for IoT. Traditional security solutions like AES, DES, RSA, and many more do not fit at all in resource constrained IoT. These traditional solutions take much power and resources from the devices part of IoT scenario. Therefore, security algorithms that are lightweight are appropriate for IoT devices due to their less power requirements and optimum memory.

Security solutions when offered with IoT will add the trust of users. This call for a secure solution requiring less power for computation and is also less vulnerable to existing attacks. The security primitives expected from IoT scenario are authentication, confidentiality as well as data integrity. In order to offer these security primitives, this research work divides the whole work in three major objectives.

First, a Hybrid Lightweight Security Framework (HLSF) is proposed. This proposed HLSF consists of three phases; first phase is registration where credentials are allocated to every device that joins the network, second phase is authentication where device and server authenticate each other so that the later communication is from the legitimate user, third phase is data security that is provided by proposed Lightweight Data Security (LDS) algorithm. Therefore, HLSF offers authentication, confidentiality and integrity services to data flowing to and from devices.

Second objective of this research work, is compare the proposed HLSF with the existing security solutions. In order to achieve this objective, first existing lightweight security solutions like SPECK, SIMON, FANTAMOS, TWINE, LDS are compared on the basis of code length, memory requirements, execution time, throughput, avalanche effect and the number of CPU cycles. This comparison shows the LDS is the better performing security algorithm among all these. Later, SPECK is compared with the proposed LDS with different variations in terms of block size, key size, code length, memory requirements, execution time, throughput, avalanche effect and the number of CPU cycles.

Third objective is to perform cryptanalysis of SPECK and LDS by implementing Differential attack on both. Results show that SPECK is compromised in 19 out of 27 rounds whereas LDS is compromised in 10 out of 20 rounds. The overall throughput, packet delivery ratio, and latency of LDS are compared with SPECK. Results prove that LDS outperforms SPECK as well is less vulnerable to differential attack. Later, the decision making is made by applying data mining on the data collected from HLSF. Effectiveness of data mining is evaluated by calculating the precision, accuracy, and recall of IoT scenario using HLSF and by comparing it with IoT scenario by using CoAP as a security mechanism.

7.2 Future Scope

IoT makes the life of user comfortable by offering a network that senses, collects, and makes decision effectively. Everything when connected to network is exposed to more number of attacks. Therefore security addition in IoT network will increase the trust level of user.

This research works in this direction only to offer an efficient lightweight security solution in IoT that can do effective data collection using sensor, performs data mining for decision making, and is less vulnerable to attacks.

The future scope of this research work can be to propose an efficient method for sensing on securely collecting the data from sensors. This will further chop down the power usage of sensor node. Moreover, this research work can be implemented in different application scenarios like smart homes, health care automation, or industry driven projects.

REFERENCES

[1] C.R Schoenberger, “The internet of things”, Forbes Magazine, Mar. 2002.

[2] European Comission, Internet of Things in 2020: A roadmap for the future. 2008.

[3] R. Morabito, V. Cozzolino, AY. Ding, N. Beijar, J. Ott, “Consolidate IoT edge computing with lightweight virtualization,” IEEE Network, vol. 32, no. 1, pp. 102-111, Jan . 2018.

[4] Dr. Hong , “Internet of Things is now "In sync with real life,” 2016.

[5] D. Bandyopadhyay, J. Sen, “Internet of things: Applications and challenges in technology and standardization,” Wireless Personal Communications, vol. 58, no. 1, pp. 49-69, May. 2011.

[6] M. Centenaro, L. Vangelista, A. Zanella, M. Zorzi, “Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios,” IEEE Wireless Communications, vol. 23, no. 5, pp. 60-70, Oct. 2016.

[7] European Commission, “Internet of things. An action plan for Europe,” Commun. from Comm. to Eur. Parliam. Counc. Eur. Econ. Soc. Comm. Comm. Reg., pp. 1–12, 2009.

[8] S. Evdokimov, B. Fabian, O. Günther, L. Ivantysynova, H. Ziekow, “RFID and the internet of things: Technology, applications, and security challenges,” Foundations and Trends in Technology, Information and Operations Management, vol. 4, no. 2, pp. 105-185, May. 2011.

[9] F. Mattern F, C. Floerkemeier, “From the Internet of Computers to the Internet of Things,” In From active data management to event-based systems and more, Springer, Berlin, Heidelberg, pp. 242-259, 2010.

[10] JS. Silva, P. Zhang, T. Pering, F. Boavida,T. Hara,NC. Liebau, “People-centric internet of things,” IEEE Communications Magazine, vol. 55, no. 2, pp-18-19, Feb. 2017.

[11] A. Whitmore, A. Agarwal,L. Da Xu, “The Internet of Things—A survey of topics and trends,” Information Systems Frontiers, vol. 17, no. 2, pp. 261-274, Apr. 2015.

[12] AK. Luhach AK, SK. Dwivedi, CK. Jha, “Applying SOA to an E-commerce system and designing a logical security framework for small and medium sized E-commerce based on SOA,” In2014 IEEE International Conference on Computational Intelligence and Computing Research, pp. 1-6, Dec. 2014.

[13] Y. Wang, X. Jia, H.K. Lee, G.Y. Li, “An indoors wireless positioning system based on wireless local area network infrastructure,” In6th Int. Symp. on Satellite Navigation Technology Including Mobile Positioning & Location Services, Vol. 54, Jul. 2003.

[14] IF. Akyildi,W. Su,Y. Sankarasubramaniam,E. Cayirci, “Wireless sensor networks: a survey,” Computer networks, vol. 38, no. 4, pp. 393-422, Mar. 2002.

[15] C. Verikoukis, R. Minerva, M. Guizani, SK. Datta, Y. Chen, and HA. Muller. "Internet of Things: Part 2." IEEE Communications Magazine vol. 55, no. 2, pp. 114-115, 2017.

[16] K. Devadiga, “IEEE 802.15. 4 and the Internet of things,” Aalto University School of Science.

[17] L. Atzori,A. Iera,G. Morabito, “The internet of things: A survey,” Computer networks, vol. 54, no. 15, pp. 2787-805, Oct. 2010.

[18] A. Garcia-de-Prado,G. Ortiz,J. Boubeta-Puig, “COLLECT: COLLaborativE ConText-aware

service oriented architecture for intelligent decision-making in the Internet of Things," Expert Systems with Applications, vol. 85, pp. 231-248, Nov. 2017.

[19] D. Miorandi,S. Sicari,F. De Pellegrini,I. Chlamtac, "Internet of things: Vision, applications and research challenges," Ad hoc networks, vol. 10, no. 7, pp. 1497-1516, Sep 2012.

[20] E. Ahmed,I. Yaqoob,A. Gani, M. Imran,M. Guizani, "Internet-of-things-based smart environments: state of the art, taxonomy, and open research challenges," IEEE Wireless Communications, vol. 23, no. 5, pp. 10-16, Oct. 2016.

[21] X. Xiaohui, "Study on security problems and key technologies of the internet of things," In2013 International conference on computational and information sciences, pp. 407-410, Jun. 2013.

[22] H. Ning,H. Liu,LT. Yang, "Aggregated-proof based hierarchical authentication scheme for the internet of things," IEEE Transactions on Parallel and Distributed Systems, vol. 26, no.3, pp. 657-667, Mar. 2015.

[23] MU. Farooq,M. Waseem,A. Khairi,S. Mazhar, "A critical analysis on the security concerns of internet of things (IoT)," International Journal of Computer Applications, vol. 111, no. 7, Jan. 2015.

[24] R. Roman,P. Najera,J. Lopez, " Securing the internet of things," Computer, vol. 1, no. 9, pp. 51-58, Sept. 2011.

[25] D. Singh, "Developing an architecture: Scalability, mobility, control, and isolation on future internet services," In2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, vol. 22, pp. 1873-1877, Aug. 2013.

[26] R. Roman,J. Zhou,J. Lopez, "On the features and challenges of security and privacy in distributed internet of things," Computer Networks, vol. 57, no. 10, pp. 2266-2279, Jul. 2013.

[27] L. Li, "Study on security architecture in the Internet of Things," InProceedings of 2012 International Conference on Measurement, Information and Control, IEEE, vol. 1, pp. 374-377, May. 2012.

[28] MR. Palattella, N. Accettura, X. Vilajosana,T. Watteyne,LA. Grieco,G. Boggia,M. Dohler, " Standardized protocol stack for the internet of (important) things," IEEE communications surveys & tutorials, vol. 15, no. 3, pp. 1389-1406, Jul. 2013.

[29] LAN/MAN Standards Committee, "IEEE standard for local and metropolitan area networks-part 15.4: Low-rate wireless personal area networks (lr-wpans)," IEEE Computer Society. Jun. 2011.

[30] D. Selent, "Advanced encryption standard," Rivier Academic Journal, vol. 6, no. 2, pp. 1-4, 2010.

[31] Y. Qiu Y, M. Ma, "An authentication and key establishment scheme to enhance security for M2M in 6LoWPANs," In2015 IEEE International Conference on Communication Workshop (ICCW), IEEE, vol. 8, pp. 2671-2676, Jun. 2015.

[32] T. Tsvetkov,A. Klein, "RPL: IPv6 routing protocol for low power and lossy networks," Network. Pp. 59-66, Jul. 2011.

[33] J. Han,M. Ha,D. Kim, "Practical security analysis for the constrained node networks: Focusing on the dtls protocol," In2015 5th International Conference on the Internet of Things (IOT), IEEE, pp.

22-29, Oct. 2015.

[34] M. Brachmann,SL. Keoh,OG. Morchon,SS. Kumar, “End-to-end transport security in the IP-based internet of things,” In2012 21st International Conference on Computer Communications and Networks (ICCCN), IEEE, pp. 1-5, Jul. 2012.

[35] A. Capossele, V. Cervo, G. De Cicco, C. Petrioli, “Security as a CoAP resource: an optimized DTLS implementation for the IoT,” In2015 IEEE international conference on communications (ICC), IEEE, pp. 549-554, Jun. 2015.

[36] J. Granjal, E. Monteiro, JS. Silva, “End-to-end transport-layer security for Internet-integrated sensing applications with mutual and delegated ECC public-key authentication,” In2013 IFIP Networking Conference, IEEE, pp. 1-9, May. 2013.

[37] C. Lu, “Overview of security and privacy issues in the internet of things,” Washington University. Pp. 1-11, May. 2014.

[38] J. Granjal, E. Monteiro, and J. S. Silva, “Security for the internet of things: A survey of existing protocols and open research issues,” IEEE Commun. Surv. Tutorials, vol. 17, no. 3, pp. 1294–1312, Jan. 2015.

[39] J. Gubbi,R. Buyya,S. Marusic,M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” Future generation computer systems, vol. 29, no. 7, pp. 1645-1660, Sep. 2013.

[40] J. Bélissent, “Getting clever about smart cities: New opportunities require new business models,” Cambridge, Massachusetts, USA. Nov. 2010.

[41] M. Chernyshev, Z. Baig, O. Bello, S. Zeadally, “Internet of Things (IoT): research, simulators, and testbeds,” IEEE Internet of Things Journal, vol. 5, no. 3, pp. 1637-1647, Jun. 2018.

[42] R. Hamidouche, Z. Aliouat, AM. Gueroui, AA. Ari, and L. Louail. "Classical and bio-inspired mobility in sensor networks for IoT applications." Journal of Network and Computer Applications, Jul. 2018.

[43] T. Heer, O. Garcia-Morchon, R. Hummen, S. L. Keoh, S. S. Kumar, and K. Wehrle, “Security challenges in the IP-based Internet of Things,” Wireless Personal Communications, vol. 61, no. 3, pp. 527-542, Dec. 2011.

[44] Y. Yang, L. Wu,G. Yin,L. Li,H. Zhao, “A survey on security and privacy issues in Internet-of-Things,” IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1250-1258, Oct. 2017.

[45] M.R. Palattella, M. Dohler,A. Grieco,G. Rizzo,J. Torsner,T. Engel,L. Ladid, “Internet of things in the 5G era: Enablers, architecture, and business models,” IEEE Journal on Selected Areas in Communications, vol. 34, no. 3, pp. 510-527, Mar. 2016.

[46] I. Kim, M.K. Back, H.J. Yim, K.C. Lee, “RFID adaptor for detecting and handling data/events in Internet of things,” Indian Journal of Science and Technology, vol. 8, no. 55, pp. 140-148, Mar. 2015.

[47] P Rachida, C.H. Chao-Hsien, X. Heng, “RFID Privacy Issues in Healthcare: Exploring the Roles of Technologies and Regulations,” Journal of Information Privacy and Security, vol. 6, no. 3, pp.

3-28, Jul. 2010.

[48] D. He, S. Zeadally, “An analysis of rfid authentication schemes for internet of things in healthcare environment using elliptic curve cryptography,” IEEE internet of things journal, vol. 2, no. 1, pp. 72-83, Feb. 2015.

[49] L. Tan, N. Wang, “Future internet: The internet of things,” In2010 3rd international conference on advanced computer theory and engineering (ICACTE), IEEE, vol. 5, pp. V5-376, Aug 2010.

[50] N. Deng, “RFID technology and network construction in the internet of things,” In 2012 International Conference on Computer Science and Service System, IEEE, pp. 979-982, Aug. 2012.

[51] B. Khoo, “RFID as an Enabler of the Internet of Things: Issues of Security and Privacy,” In2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing, IEEE, pp. 709-712, Oct. 2011.

[52] X. Jia, Q. Feng, T. Fan, Q. Lei, “RFID technology and its applications in Internet of Things (IoT),” In2012 2nd international conference on consumer electronics, communications and networks (CECNet), IEEE, pp. 1282-1285, Apr. 2012.

[53] C. Alcaraz,P. Najera,J. Lopez,R. Roman, “Wireless sensor networks and the internet of things: Do we need a complete integration?,” In1st International Workshop on the Security of the Internet of Things (SecIoT’10), 2010.

[54] S. Ezdiani, I.S. Acharyya, S. Sivakumar, A. Al-Anbuky, “An IoT environment for WSN adaptive QoS,” In 2015 IEEE International Conference on Data Science and Data Intensive Systems, IEEE, pp. 586-593, Dec. 2015.

[55] J. Liu, Y. Li, M. Chen, W. Dong, D. Jin, “Software-defined internet of things for smart urban sensing,” IEEE communications magazine, vol. 53, no. 9, pp. 55-63, Sep 2015.

[56] J.A. Stankovic, “Research directions for the internet of things,” IEEE Internet of Things Journal, vol. 1, no. 1, pp. 3-9, Feb. 2014.

[57] S. Andreev, O. Galinina, A. Pyattaev, M. Gerasimenko, T. Tirronen, J. Torsner, J. Sachs, M. Dohler, Y. Koucheryavy, “Understanding the IoT connectivity landscape: a contemporary M2M radio technology roadmap,” IEEE Communications Magazine, vol. 53, no. 9, pp. 32-40, Sep. 2015.

[58] C.W. Tsai, C.F. Lai, A.V. Vasilakos, “Future Internet of Things: open issues and challenges,” Wireless Networks, vol. 20, no. 8, pp. 2201-2217, Nov. 2014.

[59] K. Gama, L. Touseau, D. Donsez, “Combining heterogeneous service technologies for building an Internet of Things middleware,” Computer Communications, vol. 35, no. 4, pp. 405-417, Feb. 2012.

[60] H. D. Chon, S. Jun, H. Jung, and S. W. An, “Using RFID for Accurate Positioning,” Jorunal Glob. Position. Syst., vol. 3, no. 1, pp. 32–39, 2005.

[61] R. Kulkarni, A. Forster, and G. Venayagamoorthy, “Computational intelligence in wireless sensor networks: A survey,” IEEE Communication Surveys Tutorials, vol. 13, no. 1, pp. 68–96, Mar.

2011.

[62] https://volansys.com/integrating-amazon-alexa-in-your-iot-solution-for-voice-enabled-touch-free-connectivity/

[63] F. Costantino, G. Di Gravio, A. Shaban, and M. Tronci, “The impact of information sharing and inventory control coordination on supply chain performances,” Comput. Ind. Eng., vol. 76, pp. 292–306, Oct. 2014.

[64] A. Zanella, N. Bui, a Castellani, L. Vangelista, and M. Zorzi, “Internet of Things for Smart Cities,” IEEE Internet Things J., vol. 1, no. 1, pp. 22–32, Feb. 2014.

[65] L. Da Xu, W. He, and S. Li, “Internet of things in industries: A survey,” IEEE Trans. Ind. Informatics, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[66] F. H. Bijarbooneh, W. Du, E.C Ngai, X. Fu, J. Liu, “Cloud-assisted data fusion and sensor selection for internet of things,” IEEE Internet of Things Journal, vol. 3, no. 3, pp. 257-268, Jun 2016.

[67] A. Rajeswari, R. Manavalan, “Data collection methods in wireless sensor network: A study,” Int. J. Res. Appl. Sci. Eng. Technol, pp. 259-282, Sep. 2014.

[68] S. Dubey, C. Agrawal, “A survey of data collection techniques in wireless sensor network,” International Journal of Advances in Engineering & Technology, vol. 6, no. 4, pp. 1664, Sep. 2013.

[69] F. V. Meca, J. H. Ziegeldorf, P. M Sanchez, O. G. Morchon, S. S Kumar, S.L Keoh, “HIP security architecture for the IP-based internet of things,” In2013 27th International Conference on Advanced Information Networking and Applications Workshops, IEEE, pp. 1331-1336, Mar. 2013.

[70] K. Lampropoulos, S. Denazis, “Identity management directions in future internet,” IEEE Communications Magazine, vol. 49, no. 12, pp. 74-83, Dec. 2011.

[71] W. Mao, “Modern cryptography: theory and practice,” Pearson Education India, 2003.

[72] B. Schneier, “Applied cryptography: protocols, algorithms, and source code in C,” john wiley & sons, 2007.

[73] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, “Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges,” 2012 10th Int. Conf. Front. Inf. Technol., pp. 257–260, Dec. 2012.

[74] M. Wu, T. J. Lu, F. Y. Ling, J. Sun, H. Y. Du, “Research on the architecture of Internet of Things,” In2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), IEEE, vol. 5, pp. V5-484, Aug. 2010.

[75] R.H. Weber, “Internet of Things – New security and privacy challenges,” Computer law & security review, vol. 26, no. 1, pp. 23–30, Jan. 2010.

[76] B. Zhang, X. X. Ma, Z. G. Qin, “Security architecture on the trusting internet of things,” Journal of Electronic Science and Technology, vol. 9, no. 4, pp. 364-367, Dec. 2011.

[77] S. Sicari, C. Cappiello, F. De Pellegrini, D. Miorandi, and A. Coen-Porisini, “A security-and

quality-aware system architecture for Internet of Things," Information Systems Frontiers., vol. 18, no. 4, pp. 665-667–13, Aug. 2014.

[78] T. Kovacshazy, G. Wacha, T. Daboczi, C. Erdos, and A. Szarvas, "System architecture for Internet of Things with the extensive use of embedded virtualization," 4th IEEE Int. Conf. Cogn. Infocommunications, CogInfoCom 2013 - Proc., pp. 549–554, 2013.

[79] D. Zhang, L. T. Yang, and H. Huang, "Searching in Internet of Things: Vision and challenges," In2011 IEEE Ninth International Symposium on Parallel and Distributed Processing with Applications, pp. 201–206, May. 2011.

[80] Y.K. Chen, "Challenges and opportunities of internet of things," 17th Asia South Pacific Des. Autom. Conf., pp. 383–388, Jan. 2012.

[81] S. S. Basu, S. Tripathy, and A. R. Chowdhury, "Design challenges and security issues in the Internet of Things," 2015 IEEE Region 10 Symposium, pp. 90–93, May. 2015.

[82] D. Partynski and S. G. M. Koo, "Integration of Smart Sensor Networks into Internet of Things: Challenges and Applications," In2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, pp. 1162–1167, Aug. 2013.

[83] R. Billure, V. M. Tayur, and V. Mahesh, "Internet of Things- A Study on the Security Challenges," In2015 IEEE International Advance Computing Conference (IACC), pp. 247–252, Jun. 2015.

[84] Y. Yang, H. Peng, L Li, and X. Niu. "General theory of security and a study case in internet of things." IEEE Internet of Things Journal, vol. 4, no. 2, pp. 592-600, Apr. 2017.

[85] M. M. Hossain, M. Fotouhi, and R. Hasan, "Towards an Analysis of Security Issues, Challenges, and Open Problems in the Internet of Things In2015 IEEE World Congress on Services, pp. 21–28, Jun. 2015.

[86] M. Abomhara and G. M. Kien, "Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks," Journal of Cyber Security and Mobility, vol. 4, no. 1, pp. 65–88, Jan. 2015.

[87] M. J. Covington and R. Carskadden, "Threat Implications of the Internet of Things," In2013 5th International Conference on Cyber Conflict (CYCON 2013), pp. 1–12, Jun. 2013.

[88] H. Hellaoui, M. Koudil, A. Bouabdallah, "Energy-efficient mechanisms in security of the internet of things: A survey," Computer Networks, pp. 173-189, Nov. 2017.

[89] D. He, R. Ye, S. Chan, M. Guizani, and Y. Xu. "Privacy in the Internet of Things for Smart Healthcare," IEEE Communications Magazine, vol. 56, no. 4, pp. 38-44, Apr. 2018.

[90] W. Li, H. Song, and F. Zeng. "Policy-based secure and trustworthy sensing for internet of things in smart cities," IEEE Internet of Things Journal, vol. 5, no. 2, pp. 716-723, Apr. 2018.

[91] Y. Kawamoto, H. Nishiyama, N. Kato, Y. Shimizu, A. Takahara, and T. Jiang. "Effectively collecting data for the location-based authentication in Internet of Things," IEEE Systems Journal, vol. 11, no. 3, pp. 1403-1411, Sep. 2017.

[92] W. Feng, Y. Qin, S. Zhao, and D. Feng. "AAoT: Lightweight attestation and authentication of low-resource things in IoT and CPS," Computer Networks, vol. 134, pp. 167-182, Apr. 2018.

[93] O. Ruan, Y. Zhang, M. Zhang, J. Zhou, and L. Harn. "After-the-fact leakage-resilient identity-based authenticated key exchange," IEEE Systems Journal, vol. 12, no. 2, pp. 2017-2026, Jun. 2018.

[94] S. Huda, J. Yearwood, M. M. Hassan, and A. Almogren. "Securing the operations in SCADA-IoT platform based industrial control system using ensemble of deep belief networks," Applied Soft Computing, vol. 71, pp. 66-77, Oct. 2018.

[95] K.R. Choo, S. Gritzalis, and J.H. Park. "Cryptographic Solutions for Industrial Internet-of-Things: Research Challenges and Opportunities." IEEE Transactions on Industrial Informatics, vol. 14, no. 8, pp. 3567-3569, Aug. 2018.

[96] M. Ebrahim, S. Khan, and U.B. Khalid, “Symmetric Algorithm Survey: A Comparative Analysis,” International Journal of Computer Application, vol. 61, no. 20, pp. 975–8887, May. 2013.

[97] S. Cirani, G. Ferrari, and L. Veltri, “Enforcing security mechanisms in the IP-based internet of things: An algorithmic overview,” Algorithms, vol. 6, no. 2, pp. 197–226, Jun. 2013.

[98] X. Zhou and X. Tang, “Research and implementation of RSA algorithm for encryption and decryption,” In Proceedings of 2011 6th International Forum on Strategic Technology, IEEE, vol. 2, pp. 1118–1121, Aug. 2011.

[99] R. S. Jamgekar and G. S. Joshi, “File Encryption and Decryption Using Secure RSA,” International Journal of Emerging Science and Engineering (IJESE), vol. 1, no. 4, pp. 11–14, Feb. 2013.

[100] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann, and L. Uhsadel, “A Survey of Lightweight-Cryptography Implementations,” IEEE Design & Test of Computers, vol. 24, no. 6, pp. 522–533, Nov. 2007.

[101] J. Ayuso, L. Marin, A. Jara, and A. Skarmeta, “Optimization of Public Key Cryptography (RSA and ECC) for 16-bits Devices based on 6LoWPAN,” In1st International Workshop on the Security of the Internet of Things, Tokyo, Japan, pp. 1-8, 2010.

[102] Z. Liu, J. Groschadl, Z. Hu, K. Järvinen, H. Wang, and I. Verbauwhede. "Elliptic curve cryptography with efficiently computable endomorphisms and its hardware implementations for the internet of things," IEEE Transactions on Computers, vol. 66, no. 5, pp. 773-785, May. 2017.

[103] C. Cheng, R. Lu, A. Petzoldt, and T. Takagi. "Securing the Internet of Things in a quantum world," IEEE Communications Magazine, vol. 55, no. 2, pp. 116-120, Feb. 2017.

[104] L.P. Ledwaba, G.P. Hancke, H.S. Venter, and S.J. Isaac. "Performance costs of software cryptography in Securing New-Generation Internet of Energy Endpoint Devices," IEEE Access, vol. 6, pp. 9303-9323, 2018.

[105] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, “Strong Authentication for RFID Systems Using the AES Algorithm,” In International Workshop on Cryptographic Hardware and Embedded Systems, vol. 3156, pp. 357–370, Aug. 2004.

[106] P. Derbez and P.A. Fouque, “Exhausting Demirci-Sel{ç}uk Meet-in-the-Middle attacks against reduced-round AES,” In International Workshop on Fast Software Encryption, pp. 541–560, Mar. 2014.

[107] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim, and S. Chee, “HIGHT: A New Block Cipher Suitable for Low-Resource Device,” In International Workshop on Cryptographic Hardware and Embedded Systems, Springer, vol. 4249, pp. 46–59, Oct. 2006.

[108] J. Lee and D. Lim, “Parallel Architecture for High-Speed Block Cipher , HIGHT,” International Journal of Security and Its Applications, vol. 8, no. 2, pp. 59–66, Mar. 2014.

[109] D. J. Wheeler and R. M. Needham, “TEA, a tiny encryption algorithm,” In International Workshop on Fast Software Encryption, vol. 1008, no. 3, pp. 363–366, Dec. 1994.

[110] G. Sekar, N. Mouha, V. Velichkov, and B. Preneel, “Meet-in-the-middle attacks on reduced-round XTEA,” In Cryptographers’ Track at the RSA Conference, vol. 6558, pp. 250–267, Feb. 2011.

[111] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. . Robshaw, Y. Seurin, and C. Vikkelsoe, “PRESENT : An Ultra-Lightweight Block Cipher,” In International Workshop on Cryptographic Hardware and Embedded Systems Springer Berlin Heidelb, pp. 450–466, Sep. 2007.

[112] C. Blondeau, K. Nyberg, “Links Between Truncated Differential and Multidimensional Linear Properties of Block Ciphers and Underlying Attack Complexities,” In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 165-182, May. 2014.

[113] R. L. Rivest, “1 A Parameterized Family of Encryption Algorithms,” Philosophy, vol. 2., 1995.

[114] A. Biryukov and E. Kushilevitz, “Improved cryptanalysis of RC5,” In International Conference on the Theory and Applications of Cryptographic Techniques, vol. 1403, pp. 85–99, May. 1998.

[115] R. Beaulieu, S.T. Clark, D. Shors, B. Weeks, J. Smith, and L. Wingers. "The SIMON and SPECK lightweight block ciphers," In2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), pp. 1-6. Jun. 2015.

[116] A.V Duka, and B. Genge. "Implementation of SIMON and SPECK lightweight block ciphers on programmable logic controllers," In2017 5th International Symposium on Digital Forensic and Security (ISDFS), pp. 1-6. IEEE, Apr. 2017.

[117] T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi, “TWINE: A Lightweight, Versatile Block Cipher,” In ECRYPT Workshop on Lightweight Cryptography, vol. 2011, Nov. 2011.

[118] V. Grosso, G. Leurent, F.X. Standaert, K. Varıcı, “LS-Designs: Bitslice Encryption for Efficient Masked Software Implementations,” In International Workshop on Fast Software Encryption, pp. 18-37, Mar. 2014.

[119] D. J. Bernstein, “The Salsa20 Family of Stream Ciphers,” In New stream cipher designs, Springer, pp. 84-97, 2008.

[120] M.A. Alsheikh, S. Lin, D. Niyato, and H.P. Tan, “Machine learning in wireless sensor networks:

Algorithms, strategies, and applications," IEEE Communication Surveys and Tutorials, vol. 16, no. 4, pp. 1996–2018, Apr. 2014.

[121] F. Ganz, D. Puschmann, P. Barnaghi, and F. Carrez, "A practical evaluation of information processing and abstraction techniques for the Internet of Things," IEEE Internet of Things journal, vol. 2, no. 4, pp. 340–354, Aug. 2015.

[122] S. Bin, L. Yuan, W. Xiaoyi, "Research on data mining models for the internet of things," In2010 International Conference on Image Analysis and Signal Processing, IEEE, pp. 127-132, Apr. 2010.

[123] F. Chen, P. Deng, J. Wan, D. Zhang, A. V. Vasilakos, and X. Rong, "Data mining for the internet of things: literature review and challenges," International Journal of Distributed Sensor Networks, vol. 11, no. 8, pp. 431047, Aug. 2015.

[124] M. Conti, S. K. Das, C. Bisdikian, M. Kumar, L. M. Ni, A. Passarella, G. Roussos, G. Tröster, G. Tsudik, and F. Zambonelli, "Looking ahead in pervasive computing: Challenges and opportunities in the era of cyber–physical convergence," Pervasive and Mobile Computing, vol. 8, no. 1, pp. 2-21, Feb. 2012.

[125] E. W. Ngai, L. Xiu, and D. C. Chau, "Application of data mining techniques in customer relationship management: A literature review and classification," Expert systems with applications, vol. 36, no. 2, pp. 2592-2602, Mar. 2009.

[126] J. Han, J. Pei, M. Kamber, "Data mining: concepts and techniques," Elsevier, Jun. 2011.

[127] A. Sehgal, "Using the Contiki COOJA Simulator," Computer Science, Jacobs University Bremen Campus Ring, vol. 1, pp. 28759, Oct. 2013.

[128] A. Dunkels, "Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors," In 29th annual IEEE international conference on local computer networks, IEEE, vol. 16, pp. 455-462, Nov. 2004.

[129] F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with COOJA," In Proceedings. 2006 31st IEEE Conference on Local Computer Networks, pp. 641–648, Nov. 2006.

[130] Z. Ling, J. Luo, Y. Xu, C. Gao, K. Wu, and X. Fu. "Security vulnerabilities of internet of things: A case study of the smart plug system." IEEE Internet of Things Journal, vol. 4, no. 6, pp. 1899-1909, Dec. 2017.

[131] Z. Shelby, K. Hartke, C. Bormann, "The constrained application protocol (coap)," Tech. Rep.; 2014.

[132] J. Misic, M. Z. Ali, and V. B. Mišić, "Architecture for IoT Domain With CoAP Observe Feature," IEEE Internet of Things Journal, vol. 5, no. 2, pp. 1196-1205, Apr. 2018.

[133] J. Misisc, and V.B. Mišić, "Proxy cache maintenance using multicasting in CoAP IoT domains," IEEE Internet of Things Journal, vol. 5, no. 3, pp. 1967-1976, Jun. 2018.

[134] N. Correia, D. Sacramento, and G. Schütz, "Dynamic aggregation and scheduling in CoAP/observe-based wireless sensor networks," IEEE Internet of Things Journal, vol. 3, no. 6, pp. 923-936, Dec. 2016.

[135] A. Betzler, C. Gomez, I. Demirkol, and J. Paradells, "CoAP congestion control for the internet of things," IEEE Communications Magazine, vol. 54, no. 7, pp. 154-160, Jul. 2016.

[136] S.C. Son, N.W. Kim, B.T. Lee, C.H. Cho, and J.W. Chong, "A time synchronization technique for coap-based home automation systems," IEEE Transactions on Consumer Electronics, vol. 62, no. 1, pp. 10-16, Feb. 2016.

[137] C.S. Park, and W.S. Park, "A Group-Oriented DTLS Handshake for Secure IoT Applications," IEEE Transactions on Automation Science and Engineering, vol. 99, pp. 1-10, Aug. 2018.

[138] M. Vucinic, B. Tourancheau, F. Rousseau, A. Duda, L. Damon, "OSCAR: Object Security Architecture for the Internet of Things," Ad Hoc Networks, vol. 32, pp. 3-16, Sep. 2015. \

[139] C.W. Tsai, C.F. Lai, M.C. Chiang, L.T.Yang, "Data mining for internet of things: A survey, "IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 77-97, Feb. 2014.

[140] S. Sun, "Analysis and acceleration of data mining algorithms on high performance reconfigurable computing platforms," 2011.

[141] G. Kesavaraj and S. Sukumaran, "A study on classification techniques in data mining," In2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), pp. 1–7, Jul. 2013.

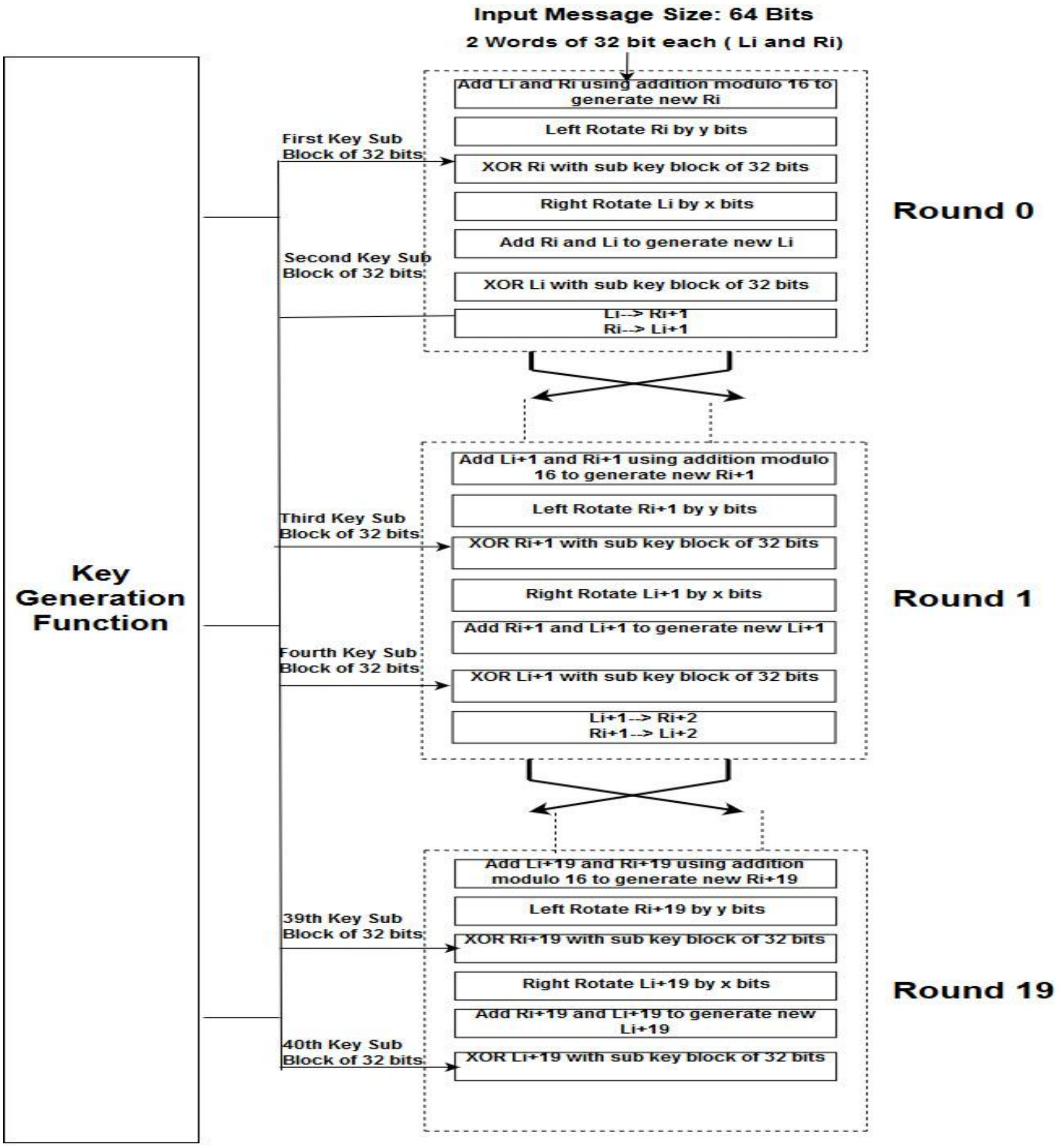
[142] E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems," In Journal of CRYPTOLOGY, vol. 4, no. 1, pp. 2–21, Jan. 1991.

[143] I. Dinur, "Improved Differential Cryptanalysis of Round-Reduced Speck," In International Workshop on Selected Areas in Cryptography, Springer, pp. 147-164, Aug. 2014.

APPENDIX I

FULL DIFFUSION IN LDS

The full diffusion of LDS is shown in this section using 20 rounds of operation. LDS takes input in the form of 64 bit. The working of LDS as shown in Figure 3.1 is reproduced in this section to justify the process of diffusion in LDS.


Input Message Size: 64 Bits
2 Words of 32 bit each (Li and Ri)
Key Generation Function
First Key Sub Block of 32 bits
Second Key Sub Block of 32 bits
Add Li and Ri using addition modulo 16 to generate new Ri
Left Rotate Ri by y bits
XOR Ri with sub key block of 32 bits
Right Rotate Li by x bits
Add Ri and Li to generate new Li
XOR Li with sub key block of 32 bits
Li--> Ri+1
Ri--> Li+1
Round 0
Third Key Sub Block of 32 bits
Fourth Key Sub Block of 32 bits
Add Li+1 and Ri+1 using addition modulo 16 to generate new Ri+1
Left Rotate Ri+1 by y bits
XOR Ri+1 with sub key block of 32 bits
Right Rotate Li+1 by x bits
Add Ri+1 and Li+1 to generate new Li+1
XOR Li+1 with sub key block of 32 bits
Li+1--> Ri+2
Ri+1--> Li+2
Round 1
39th Key Sub Block of 32 bits
40th Key Sub Block of 32 bits
Add Li+19 and Ri+19 using addition modulo 16 to generate new Ri+19
Left Rotate Ri+19 by y bits
XOR Ri+19 with sub key block of 32 bits
Right Rotate Li+19 by x bits
Add Ri+19 and Li+19 to generate new Li+19
XOR Li+19 with sub key block of 32 bits
Round 19

This structure of LDS represents that the input is processed in two portions left and right of 32 bit each. Each block is processed using a key generated from the mixing function. In every round, each half is processed using addition, XOR, and rotations. Full diffusion of LDS will assure that 20 rounds will lead to propagation of change in all the bits by making change in the single bit of input. It is assumed in the example the last bit that is the bit no 32 is changed from the left sub block.

In 1st Round, L_0 that is, left sub block is first shifted by 7 bits. Therefore the changed bit no 32 will make changes in bit no 25. For Right block, the left is added with right thus affecting the bit no 25 and 32 of right block also and finally the shifting of right block by 3 bits will affect the bit no 3 of the right sub block. Therefore, in 1st round, total five bits are affected by changing a single bit. Two are affected bit is in left block and two are affected in right block.

In 2nd Round, the output received from right block is L_1 now and output received from left sub block in round 1 is R_1. Rotation is applied by 7 bits on L_1 that will affect bit no 28 of left sub block. For Right block, the left is added with right thus affecting the bit no 28 of right block also and finally the shifting of right block by 3 bits will affect the bit no 31 of the right sub block. Therefore in 2nd round, total three bits are affected by changing a single bit. One affected bit is in left block and two are affected in right block.

In 3rdRound, the output received from right block is L_2 now and output received from left sub block in round 1 is R_2. Rotation is applied by 7 bits on L_2 that will affect bit no 24 of left sub block. For Right block, the left is added with right thus affecting the bit no 24 of right block also and finally the shifting of right block by 3 bits will affect the bit no 27 of the right sub block. Therefore in 3rd round, total three bits are affected by changing a single bit. One affected bit is in left block and two are affected in right block.

In 4thRound, the output received from right block is L_3 now and output received from left sub block in round 1 is R_3. Rotation is applied by 7 bits on L_3 that will affect bit no 20 of left sub block. For Right block, the left is added with right thus affecting the bit no 20 of right block also and finally the shifting of right block by 3 bits will affect the bit no 23 of the right sub block. Therefore in 4th roundtotal three bits are affected by changing a single bit. One affected bit is in left block and two are affected in right block.

Therefore it can be noticed that in first round 5 bits were affected by changing 1 bit and in rounds from 2 to 19, 3 bits are affected by changing only one bit in the original text. In last round 5 bits are affected due to final addition operation. As there are total 64 bits in original text. Therefore, minimum 20 rounds are required to perform full diffusion in all the 64 bits by changing a single bit in input.