

A FUZZY BASED METHODOLOGY TO IMPLEMENT SECURITY IN INTERNET OF THINGS

A Dissertation submitted in partial fulfillment of the requirement for
the award of degree of

Master of Technology

In

Software Engineering

Submitted by

Aanchal Punia

(Roll No.- 2K15/SWE/01)

Under the guidance of

Prof. (Dr.) Daya Gupta



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

BAWANA ROAD, DELHI

2015-2017



DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

CERTIFICATE

Date: 28/06/17

This is to certify that the work embodied in the thesis entitled "A Fuzzy based methodology to implement security in Internet of Things" submitted by **Aaschal Punia** with Roll no. **2K15/SWE/01** as a *full-time* research scholar in the Department of Computer Science and Engineering, Delhi Technological University, is an authentic work carried out by her under my guidance and is submitted to Delhi Technological University for the award of the Degree of **Master of Technology**.

This work is original research and has not been submitted, in part or full, to any other University or Institute for the award of any degree.

Supervisor

A handwritten signature in blue ink, appearing to read 'Daya Gupta', is written over the printed name.

Dr. Daya Gupta

Professor,

Department of Computer Engineering

Delhi Technological University

Delhi-110042

SHAHBAD DAULATPUR, BAWANA ROAD, DELHI-110042, INDIA

OFF: +91-11-27871018 FAX: +91-11-27871023 WEBSITE: www.dtu.ac.in

ACKNOWLEDGEMENT

I am very thankful to **Dr. Daya Gupta** (Professor, Computer Science and Engg. Dept.) and all the faculty members of the Computer Science Engineering Dept. of DTU. They all provided us with immense support and guidance for the project.

I would also like to express gratitude to **Ms. Shruti Jaiswal** (Research Scholar, Delhi Technological University) for providing me continuous support and guidance during this project. Her presence throughout this process provided me the motivation and right directions to complete my work.

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

I would also like to thank my flatmates Jagriti Singh, Monika Singh, Meenakshi Dayal, Sonia Tomer and Juli Keshari.

Last but not the least, I would like to thank my parents, Prof. Rajkala and Dr. Kuldeep Singh Punia, brother Anurag, sister Dr. Aditi and close friends Pankaj Choudhary, Shefali Modi, Swati Singh, Indu Yadav and Parul Hooda for their support, love, care and sacrifices for educating and preparing me for my future endeavor.

Aanchal Punia

Roll No. 2K15/SWE/01

Abstract

Internet has become a necessity today because of the technological advancement. With the needs of connecting everything and anything, Internet of Things (IoT) comes to the picture. IoT can be defined as "a dynamic global network infrastructure." IoT has changed the way we think, live and work. IoT is making the whole world a connected web. One can control, track and manage the things/objects/business by sitting miles away. Also, data can be accessed anytime and anywhere. IoT is a network of heterogeneous networks connecting together via the internet. IoT is evolving at a rapid pace.

Everything has downsides, and IoT is facing the biggest challenges of security and privacy. Thus, the exponentially growing popularity of IoT needs a security check. Any breakdown in security of IoT will put the connected objects and human lives at risk. With the increase in the size of IoT, the number of objects/things at stake will also rise.

Intercommunicating smart devices provide real-time responses and act accordingly with less or no human intervention. These talking machines make things easy but if they are not guarded well, they can create chaos. Also, IoT involves private confidential data, financial records, human lives, and ecosystems.

Attackers can exploit the loopholes in the IoT security and can harm the humans or the system either directly or indirectly. Hence, security of IoT has to be ensured. To give an estimate of the efficiency of the present day security algorithms, a security engineering framework [40] is applied to a system based on IoT. This work aims at identifying all the possible ways a system can be attacked and try to suggest optimal security algorithms which can mitigate the identified security issues. After all, "a secure IoT guarantees a safe world."

Keywords: Internet of Things (IoT), security, fuzzy logic, risk, threat prioritization, security index, security engineering framework, security requirements.

Table of Contents

Acknowledgement	iii
Abstract	iv
Table of Contents	v
List of Figures	vi
List of Tables	vii
1. Introduction.....	1
1.1 Introduction	1
1.2 Motivation	3
1.3 Related Work	4
1.4 Problem Statement	5
1.5 Scope of work & approach	6
1.6 Organization of thesis	7
2. Background.....	8
2.1 IoT Architecture	8
2.2 IoT Security Issues	9
2.3 Security Requirements	11
2.4 Available Security Techniques	14
2.5 Security Engineering	23
3. Security Requirements Engineering for IoT.....	25
3.1 Modified Security Engineering Framework	25
3.2 Security Requirements Engineering	26
4. Security Design Engineering for IoT.....	45
4.1 Security Design Engineering	45
5. Case Study.....	52
5.1 Smart Home Model	52
5.2 Security Requirements Engineering	56
5.3 Security Design Engineering	58
6. Tool.....	60
6.1 Introduction	60
6.2 Working of Security Engineering Framework for IoT	61
7. Conclusions and Future Work.....	68
7.1 Conclusions	68
7.2 Future Work	69
References.....	70

List of figures

Figure No.	Caption	Page No.
Figure 2.1	IoT architecture: Layers and their protocols.....	8
Figure 2.2	Security Engineering Framework.....	24
Figure 3.1	Modified Security Engineering Framework for IoT.....	26
Figure 3.2	Steps in Requirements Elicitation.....	27
Figure 3.3	Fuzzy system for calculating risk.....	40
Figure 3.4	Membership Function: Threat Rating.....	40
Figure 3.5	Membership Function: Impact Rating.....	41
Figure 3.6	Membership Function: Risk.....	42
Figure 3.7	Defuzzification: Finding value of Risk for input [25;10].....	43
Figure 4.1	Steps of Security Design Engineering Phase.....	45
Figure 5.1	Communication and information flow in smart home network.....	53
Figure 5.2	Overall architecture of Smart Home automation.....	55
Figure 6.1	Home screen of Tool.....	60
Figure 6.2	Second screen with tabs for each phase.....	61
Figure 6.3	Tab 1- Requirements Elicitation.....	62
Figure 6.4	Tab 2-Requirement Analysis.....	63
Figure 6.5	Tab 3-Risk Calculation.....	64
Figure 6.6	Tab 4-Threat Prioritization.....	65
Figure 6.7	Tab 5-Security Req. mapping.....	66
Figure 6.8	Tab 6-Security Index.....	67

List of Tables

Table No.	Caption	Page No.
Table 2.1	IoT layers and corresponding issues.....	11
Table 2.2	Analysis of available security techniques.....	19
Table 3.1	Assets corresponding to actors.....	29
Table 3.2	Identification of vulnerabilities based on actors.....	30
Table 3.3	Threat-Vulnerability mapping.....	31
Table 3.4	Security requirements-Threats Mapping.....	34
Table 3.5	Asset and actor mapping (Asset Rating).....	35
Table 3.6	Vulnerability and actor mapping (Vulnerability Rating).....	36
Table 3.7	Threat - Asset Mapping (Impact rating).....	37
Table 3.8	Threat and Vulnerability (Threat Rating).....	38
Table 3.9	Weights of Threat Rating Variable.....	40
Table 3.10	Weights of Impact Rating Variable.....	41
Table 3.11	Weights of Risk variable.....	41
Table 3.12	Rules for the Fuzzy system.....	42
Table 3.13	Calculation of Risk for threats.....	43
Table 3.14	Prioritization of threats.....	44
Table 4.1	Risk values corresponding to Security requirements.....	46
Table 4.2	Security Requirements and Security Services mapping.....	48
Table 4.3	Security Algorithms Pool.....	49
Table 4.4	Security Requirements and Algorithms mapping.....	51
Table 5.1	Layer wise components of smart home.....	54
Table 5.2	Actors for Smart Home.....	56
Table 5.3	Assets for Smart Home.....	56
Table 5.4	Risk values for Security Requirements.....	58
Table 5.5	IoT constraints and their specifications for security algorithms.....	59
Table 5.6	Security Index of Algorithms.....	59

This chapter provides insights into the internet of things technology. The history of IoT and its applications are also discussed. It also highlights the importance of security in IoT. The problem statement is stated along with the reasons that lead to the motivation for working in this field.

1.1 Introduction

Internet of things can be defined as "a network of computing devices and objects embedded with electronic chips communicating via the internet and performing the tasks of sensing, actuation and data transfer." Internet of Things is abbreviated as IoT. IoT aims at connecting each and every machine and device to provide a ubiquitous connectivity. This ubiquitous connectivity means one can control, access, manage, sense and actuate any remote system which is connected to the internet. Internet which has previously connected the clients and the servers has widened up its scope and is now connecting everyday objects to one another. This new addition of objects and their actuation to the traditional internet is referred to as Internet of things. IoT connects the secluded sensor networks, home networks, industrial networks and much more. IoT provides seamless connectivity and remote access. Besides M2M (machine to machine) communications, it is also making M2H (machine to human) and H2M (human to machine) communications a reality.

IoT can also be defined as a collection of things which can interact and cooperate among themselves to reach common goals. Things in IoT can range from smart devices like smartphones, web apps to passive devices like RFID tags, temperature sensors, etc. Things/objects which have unique addresses and are connected to the internet can become a part of IoT. This interconnection among the devices is possible because of the embedded electronics and computational abilities. The purpose of IoT is to provide a 24*7 control and actuation of a real life task with minimal human intervention. Things in IoT are called as "nodes". Every node has some role assigned to it. Each node behaves as per the roles assigned to it. A node can collect, send and act on data. The data is acquired from the surrounding environments. From the above discussion we can conclude that "IoT is capable of changing the lifestyle and the work style of people."

History of IoT: The concept of IoT was firstly proposed by Kevin Ashton, executive director of the Auto-ID Center in 1999. He referred the IoT as "uniquely identifiable interoperable connected objects with radio-frequency identification (RFID) technology." Neil Gershenfeld also wrote something about this in his book "When Things Start to Think". In 2005, IoT was acknowledged by UN's International Telecommunication Union (ITU). EU (European Union) also recognized IoT, and the first European IoT conference was held in 2007. The Internet of Things was born in between 2008 and 2009 as estimated by Cisco Internet Business Solutions Group (IBSG) since this was the time when the count of "things or objects" connected to the internet exceeds that of the people connected. The wide use of easily available devices which can connect to the internet is one of the reasons behind this increase.

IoT applications: IoT applications are ranging from intelligent transportation to smart traffic, smart homes and smart grid to intelligent urban management. Also, it covers the market, health, logistics, surveillance, industry, and agriculture. It seems as if no aspect of day to day life is left untouched by IoT. Initial IoT projects comprise of a water fountain whose height and flow imitated the trends of the stock market [1]. Some of the IoT applications are smart homes, wearables, smart city, smart grid, industrial internet, connected vehicles, smart health monitoring, vehicles, smart retail, transportation and logistics, smart supply chain, agriculture, traffic management, and control.

This wide range of applications means the involved data may be personal, professional, medical, sensitive or financial. Also, in today's information age, information is an asset. That is why information needs to be handled carefully and prevented from attacks. Information has to be hidden from unauthorized access (confidentiality), protected from unauthorized change (Integrity) and available to an authorized entity when it is needed (availability). CIA (confidentiality, integrity, availability) is to be ensured for a system to be safe and secure [44]. Insecure IoT leads to unauthorized access, the disclosure of personal information, leakage of top security data and corruption of data. Also, the stakes are high in case of a security breach. There are currently about 6.4 billion IoT devices deployed worldwide, with a rise to 20.8 billion by 2020, according to Gartner [2]. Henceforth, security of IoT is of greater significance because of this large-scale involvement of people and things. Also, any security attack can directly affect the humans as almost every IoT application like smart home, smart city, smart grid, transport and vehicle control, etc. involves human.

1.2 Motivation

IoT is surprisingly growing at a rapid pace, leaving behind the notion that IoT might end up just being a dream. One can see IoT applications in the field of real-time location services (connected cars, supply chain automation, and asset tracking), industrial automation and facilities management, energy and utilities (smart grid, smart metering) and public safety and emergency services (early flood warning). It means IoT has a direct impact on human life, environment, and property, hence any threat to IoT will directly harm the humans and the environments involved. Therefore, security of IoT systems has to be ensured. Security is important as it protects the information and the system against any internal or external harm or damage. Besides being a collection of heterogeneous networks, IoT also consists of devices with different power, memory, and computational constraints. These devices, as mentioned in the previous section can be web interfaces, RFID tags, Android applications, embedded systems, etc. These devices and networks, when working independently have efficient and effective security techniques which ensure their safe and secure deployment. But when brought together, this fusion can pose new challenges and threats to the security. This implies that "security of IoT" is a domain with ample research opportunities where new security techniques, frameworks, and methodologies can be proposed to ensure full security and safety of IoT environments. Also, implementing security in IoT is a complex task because of the device constraints. Hence, this research area is chosen for the thesis.

Smart homes, an IoT-based system is gaining popularity among the tech giants like Google, Amazon, Logitech, LG and the startups as well. They aim at providing systems which can automate the mundane tasks of switching the lights, locking the doors, home surveillance, etc. and keeps the user connected to his home even if he is miles away. The literature survey of home automation [3][4] indicates that because of different manufacturers and service providers, providing security to such a heterogeneous collection of devices is a complex task. Therefore, the motive of this work is to come out with an efficient and innovative structured approach which can provide a high-level security to the smart homes by taking care of all the security requirements. The proposed framework considers the device constraints as well.

1.3 Related Work

Research in the field of IoT has identified both the general and security issues in the IoT domain. The research in the IoT security field includes survey papers, implementations, proposed algorithms, security frameworks and secure IoT-based systems. This section presents a brief review of the related work which is explained in detail in the next chapter.

In [5], researchers have presented a survey of technologies, applications and research challenges for IoT. The vision for IoT and definition of smart objects is introduced. Support for heterogeneous devices, need of equipping sensors with a battery and dimensions of electronics that are to be embedded in IoT objects; are the three main limiting factors identified by the authors. The work also highlights that the key concepts from SOA (service-oriented architecture) can be exploited to find an optimal and lightweight solution for IoT. The identified research areas are: Distributed intelligence; Distributed systems; and Security (data confidentiality, privacy, and trust).

In [6][7], researchers have identified the security issues specific to the layers in IoT architecture. In [8], a systematic approach which solves IoT security issues is proposed.

For confidentiality, cryptographic algorithms are explored. Kakali et al. [17] proposed an authentication based scheme on the basis of Elliptic Curve Diffie-Hellman (ECDH). In [11], Thomas et.al included Datagram Transport Layer Security (DTLS) handshake in proposing a two-way authentication security scheme. A dynamic variable cipher security certificate [12], standard compliance framework for IEEE 802.15.4 [19], identity based encryptions [14] [15], public key infrastructure (PKI) [15] are some of the schemes proposed to make the IoT-based system secure against denial and man in the middle attacks.

In [35], researchers proposed a capability-based access control (CapBAC) mechanism to provide more sophisticated techniques for access control.

In [9], Antonio et al. proposed an architecture which provides secure communication, authentication, and privacy to the healthcare domain of IoT and also prevents eavesdropping and denigration of service.

In [10], Liang et al. designed a new and efficient multimedia traffic security framework which addresses the issues of key management, authentication and watermarking in the transportation domain of IoT.

In [16], Shahid et al. implemented a novel intrusion detection architecture on Contiki OS (OS for IoT). This implementation prevented the system against routing attacks.

In [45], an android application for controlling home is developed. This application allows control and remote access, but it does not talk about security of the system.

In [3][4], researchers have identified the research gaps in smart home systems. Security of smart homes is identified as the biggest issue. Therefore, home automation systems are to be secured against malicious activities.

Security engineering framework [40] suggests that if requirements are elicited and then the algorithms for implementing security are chosen, the efficiency and acceptance of the system would increase. Therefore, the thesis works towards engineering the approach to implement security in IoT-based system.

1.4 Problem statement

Security in IoT is different from security in traditional networks because IoT isn't a standalone technology, it is a combination of different technologies like Bluetooth, ZigBee, wireless networks, internet networks, cloud storage, big data and others. Therefore, for ensuring security in IoT, the security issues of the different technologies and issues arising from their combination should be taken care of. In the previous section, we have seen that implementing security in IoT is a complex task. For the identification of security issues and for implementing security requirements, a structured mechanism is required which will elicit, analyze and prioritize the security requirements.

This thesis is inspired from a structured framework proposed by Kakali et al.[40]. Hence, the problem statement of this thesis is: "**Identifying security in smart homes which is a system based on IoT.**" It emphasizes on addressing the security issues present in smart home systems based on the structured approach.

Security for a system depends on the actors, assets, vulnerabilities, threats, security requirements and security algorithms implemented. All these factors are co-related. The co-relations among them should be identified. Threats should be prioritized on the basis of risk values. Security algorithms should not be chosen in ad-hoc manner, therefore, a method for selection of efficient security algorithms is required.

1.5 Scope of work & approach

1.5.1 Scope of work

The goal of this work is to adopt the security engineering framework [40] and suggest security algorithms based on the elicitation of the security requirements for home automation based systems. This process is carried out in two phases: Security Requirements Engineering and Security Design Engineering. The first part deals with elicitation and analysis of the security requirements, threat prioritization based on associated risk. The second phase maps the security requirements and security services, creates a repository of security algorithms and calculates security index for the algorithms. The scope of this work can be summarized as:

- Requirements Engineering phase of the generic security framework is adopted for home automation system.
- Design Engineering phase is modified and the constraints specific to home automation system are considered. Based on these constraints, the method to choose an efficient algorithm is required.
- This methodology should be illustrated for smart home case study.
- A tool based on security engineering framework is developed.

1.5.2 Approach

This work aims at modifying the pre-existing security engineering framework as per the need of the IoT-based systems. The generic framework was divided into four different phases [40]: Security Requirements engineering, Security Design Engineering, Implementation and Security Testing.

In our work, we have taken into consideration the first two phases only. In the requirements phase, the actors, assets, vulnerabilities, threats and security requirements for the home automation systems are identified. Tables for asset-actor mapping, actor-vulnerability mapping, vulnerability-threat mapping, threat-security requirement mapping and other mappings are created. To prioritize threats, risk calculation is required. The risk is calculated using fuzzy logic. Threats are then prioritized based on the risk assessment result.

In the design phase, security requirements are mapped to the security services. This mapping will help in identifying the security algorithms which will implement the security services.

Security index is calculated for the algorithms in the repository. Algorithms are then selected for implementation on the basis of security index, thus preventing the selection of security algorithms on ad hoc basis. This selection also considers the environment and device constraints present in the IoT-based home automation systems.

For the better understanding of the applied framework, we have applied it on an IoT application of "**smart home automation.**" Also, a tool is developed to implement the modified security engineering framework.

1.6 Organization of Thesis

This sub heading gives brief details about the chapters in this thesis.

Chapter 2: Background- This chapter discusses IoT in detail. The chapter specifies the architecture for IoT, the concerned security issues and the security requirements. The techniques available to fulfill these requirements are explained and summarized in a table for better understanding. This chapter also gives insight about the generic security engineering framework.

Chapter 3: Security Requirements Engineering for IoT- This chapter discusses the modified security engineering framework and then tells about the first phase: security requirements engineering. It discusses the basis for identification of various system security requirements. The risk associated with each threat is calculated. Threats are then prioritized on the basis of the risk value.

Chapter 4: Security Design Engineering- This chapter discusses the second phase of the work. Security requirements are mapped to the security services. Security mechanisms are identified for these services. Based on the threats covered, security index for these algorithms is calculated.

Chapter 5: Case study- In this chapter, a smart home model which is taken as a case study is described first. Then the framework is applied to this system and the results are shown.

Chapter 6: Tool- This chapter describes the tool developed for implementing security in the smart home, an IoT-based system.

Chapter 7: Conclusions and Future Work- This chapter concludes the work done in this thesis along with the future work that can be done on the basis of this work.

This chapter discusses Internet of things and security engineering. In IoT, an architecture for IoT is defined, and on the basis of this architecture, the security issues related to each layer are identified. IoT security issues and security requirements are also discussed. This chapter gives an overview of the available techniques which are responsible for ensuring security in today's IoT. In security engineering, the generic framework proposed for ensuring security is discussed.

2.1 IoT Architecture

The architecture of IoT has not been standardized yet, unlike the traditional internet, which follows TCP/IP protocol. IoT is divided into three layers: perception layer, transportation layer, and application layer, as per the proposed architecture of ITU-T Y.2002 [2]. The layers communicate with one another. The IoT layers stack is shown in Figure 1.

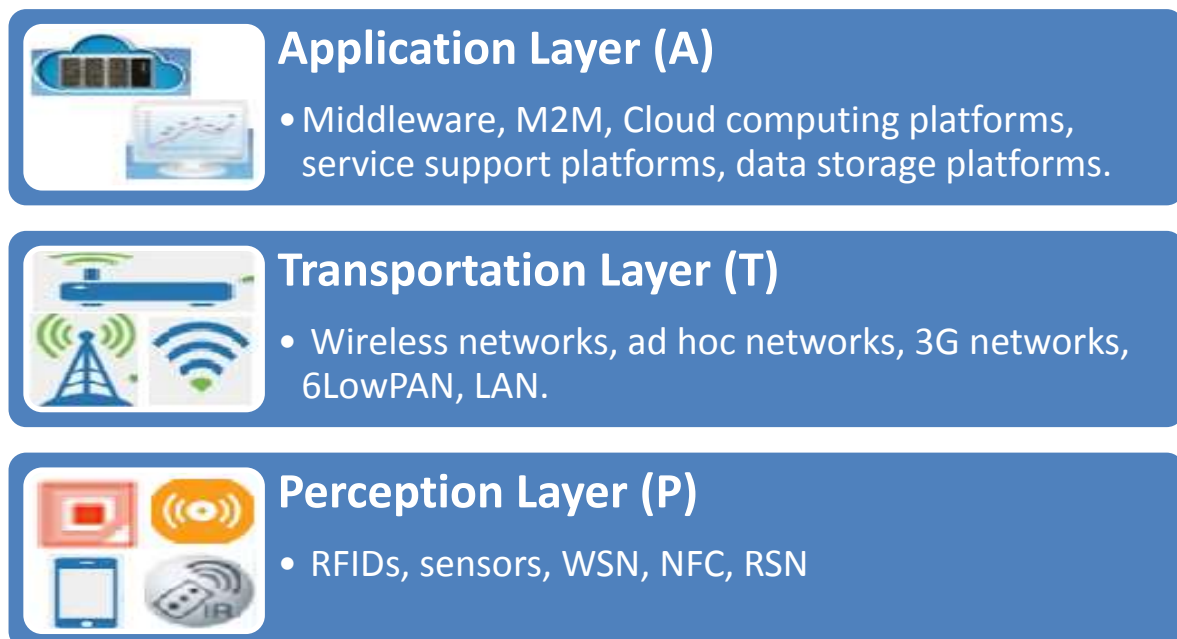


Figure 2.1: IoT architecture: Layers and their protocols

Perception layer deals with the physical aspects of IoT. It consists of sensors, controllers or RFIDs. The nodes are used for data acquisition and data control. It connects all the nodes internal to a sensor network via Wireless Sensor Networks (WSN) or Radio Sensor Networks (RSN).

Transportation layer connects the remote and heterogeneous sensor networks to the outside world. It mainly provides global access for the perception layer, the perception of information transmission and storage . It includes wireless networks (like cellular networks, wireless LAN, Wi-Fi) and ad-hoc networks for data access. For data transmission, Ipv4-based Internet, 6LowPAN and other technologies are used.

Application layer interacts directly with the end user and supports all kinds of business services. It also includes middleware, M2M, cloud computing platforms and service support platforms. It processes and analyzes the data collected from the transportation layer. It supports IoT applications such as intelligent transportation, smart home, smart healthcare, etc.

2.2 IoT Security Issues

In an IoT environment, the security concerns broadly consist of data confidentiality, privacy, and trust. A threat is a possible way by which the system can be attacked or damaged. Security issues or threats both have the potential to harm a system. Security issues in IoT are:

- **Information Leakage:** Information leakage means revealing of information to an unauthorized party. In IoT, all the layers deal with data. Therefore, an attacker can target any of the three layers to get some information which is not intended for him. In perception layer, the attacker can overhear the information by listening and recording the data generated by the sensors. In transportation layer, some of the algorithms might have a backdoor, which can leak information. At the application layer, data is processed and analyzed. This data can be accessed by the attacker with the use of malicious software like Trojans or worms.
- **Eavesdropping:** The act of secretly listening to others private conversation is eavesdropping. At perception layer, the eavesdropper can get a hold of the information by intercepting the electrical or radio signals which carry the data in WSN or RSN. At the application layer, one can target the user or business to intercept the private conversation.
- **Data modification:** Data modification indicates that integrity of the data is compromised. Data can be corrupted at the level it originates, during its transfer or at the point where it is stored. So, this issue effects all the three layers of IoT architecture.

- **Unauthorized access:** Authorization is essential to ensure that only the authorized personnel can get hold of the system. Unauthorized access to the system can lead to information leakage and system misuse. At the physical layer, an unauthorized hold of any sensor means data alteration. At transportation layer, unauthorized access to a gateway means that the information about the higher level users and traffic flow can be obtained for malicious use. At the application layer, unauthorized access can manipulate the working of a system.
- **Forgery:** It can be done at perception layer only. In this attack, an imitation of the existing node, i.e. a fake node becomes a part of the sensor network. This fake node can gain access to the system by using credentials of a legitimate node. It can misguide the system by providing false data. It can harm the system internally, as this node becomes a part of the system.
- **Phishing:** Phishing is the attempt to obtain sensitive information such as usernames, passwords, and credit card details (and sometimes, indirectly, money), often for malicious reasons, by camouflaging as a trustworthy entity in an electronic communication. The user might think that he is giving the details to the authentic party but in reality, the details are collected by an unauthorized and unknown party. It exploits the application layer protocols.
- **DDoS/DoS:** Distributed DoS or DoS (denial of service) attacks aim at flooding the network with garbage data so that the system gets overloaded and halts. It exhausts the network bandwidth at the transportation layer which leads to unavailability of applications. For a DDoS attack to happen, attacker targets the nodes at perception layer, infect them with a malware, and then these nodes unintentionally sends data traffic to the targeted website or server.
- **Conflict Collision:** RFID tags are one of the most important components of IoT. They have shorter range and requires less power, hence best suited for any small area network. But these tags have collision issues. Tags' collision and readers' collision are the two types of RFID collisions. The first collision occurs because of a large number of the tags. The later because of the overlapping scopes of the readers at the perception layer. This conflict leads to data redundancy.
- **Network Paralysis:** IoT generates a lot of data. There can also be a lot of noise because of the various microcontrollers and the microchips. This noise can interfere

with the transmitting signals and leads to network paralysis. This issue occurs in the transportation layer.

- **Routing attack:** It is the spoofing of the routing table. Transportation layer maintains the routing table. Hence this issue can occur at the transportation layer only. It causes the packets meant for an IP address to be sent to the attacker.

These security issues effect different layers of IoT architecture. Table 2.1 shows the security issues and the layers effected by these issues.

Table 2.1: IoT layers and corresponding issues

IoT Layers→ Security Issue↓	Information leakage	Limited power	Less computation	Data Security	Unauthorized Access	Forgery	Phishing	DDoS/DoS	Network paralysis	Routing Attack
Perception Layer	√	√	√	√	√	√				
Transportation Layer	√			√	√		√	√	√	√
Application Layer	√			√	√		√	√		

2.3 Security Requirements

This section describes security requirements proposed by Firesmith for an IoT-based system. Also, IoT has some additional security requirements [38] that apply to IoT systems only. Additional constraints for IoT system are also discussed here.

2.3.1 Firesmith Security Requirements

Firesmith [39] pointed out that security requirements should be treated differently from the architectural security mechanisms. The security requirements as listed by Firesmith in his work are explained for the IoT-based system.

- **Identification Requirement:** This security requirement deals with identifying the actors who can be given access to the system. Identification is important for authentication. Actors can be identified on the basis of who they say they are (name, unique id number); what they have (digital token, digital certificate) and who they actually are (face recognition, fingerprint, retina scan). User name, password, id,

digital certificates, tokens, fingerprints, face recognition, etc. are the techniques used to identify an individual. Anonymity, if required by any user, should also be provided to ensure privacy requirement fulfillment. It resolves **forgery** security issue as mentioned in section 2.2.

- **Authentication Requirement:** This security requirement deals with verifying the user once he has provided his identity details, i.e. it checks whether the user is who he says he is or not. Identification is required for authentication. Once authenticated, the user can get into the system and can perform the tasks assigned to him. The main aim of this requirement is to prevent any compromising situation where an attacker, who by modifying the identification details of any actual user, tries to get into the system. It resolves **phishing** security issue as mentioned in section 2.2.
- **Authorization Requirement:** This security requirement deals with assigning access to the users and applications as per their roles in the system. Once the user is identified and authenticated, then he can access the information or applications for which he is explicitly authorized. This requirement is important so that one user cannot modify or delete details or information of another user from the system. It also ensures confidentiality and privacy and prevents damages which can be caused by any unauthorized access. Mechanisms used to ensure this requirement are: Authorization lists, physical access controls like locks or hardware electronic key, etc. It resolves **unauthorized access** security issue as mentioned in section 2.2.
- **Immunity Requirement:** This security requirement specifies how safe the system is from any external attacks, i.e. to which extent the system can safeguard it against the unauthorized access and undesirable programs like worms, computer viruses, malware etc. Mechanisms used to ensure this requirement are: antivirus software and firewalls. It resolves **DDoS** security issue as mentioned in section 2.2.
- **Integrity Requirement:** This security requirement deals with ensuring that any unauthorized creation, modification or deletion has not corrupted the actual message. It ensures trust between the two communicating parties. Mechanisms used to ensure this requirement are: cryptography, hash codes and digital signatures and certificates. It resolves **data modification** security issue as mentioned in section 2.2.
- **Intrusion detection Requirement:** This security requirement deals with detecting and recording any suspicious activity that threatens the safety, security, and integrity of the system. It keeps a record of unsuccessful login attempts, unauthorized access,

abnormal behavior of the user or application. Mechanisms used to ensure this requirement are: alarms, event logging, intrusion detection and prevention systems.

- **Non-repudiation Requirement:** This security requirement ensures that any user cannot turn back on the communication once done by him. The main aim of this requirement is to prevent an actor from denying that he was a part of any past communication or from modifying any message once it has reached the recipient. It confirms that the records are tamper proof and no one can change or modify the records. Mechanisms used to ensure this requirement are: timestamps, digital signatures, hashing, encryption.
- **Privacy Security Requirement:** Privacy means keeping personal and sensitive data safe from the access of any unauthorized person. This security requirement takes care of how much sharing should be allowed for any system. This requirement is fulfilled in the e-marketplaces, where personal information of the customer is hidden from the merchant and the merchant is supplied with non-private information and all communications are strongly encrypted. Mechanisms used to ensure this requirement are: encryption and hashing.
- **Security Auditing Requirements:** After applying all the security mechanisms to fulfill the corresponding security requirements, an audit of the system is then required. This security requirement keeps a check on the status of these mechanisms, i.e. whether they are updated or not, whether they are working or not, whether they are disabled or enabled. Along with auditing, it also maintains the log of the system. Mechanisms used to ensure this requirement are: Audit reports, audit trails and event logs.
- **Survivability Requirement:** This requirement exhibits the resistance of the system to any malicious attack or accidental hardware/software failure and also ensures that if the system goes down then minimal or no harm should occur to the stakeholders and the users. In simple terms, it tells how well a system can survive any intentional and unintentional damage. System recoverability after any failure is not of concern here. Mechanisms used to ensure this requirement are: hardware redundancy, data center redundancy.
- **Physical Protection Requirements:** This security requirement is concerned with the physical harm done in the real world, like theft, destruction, replacement, sabotage, etc. This requirement signifies that along with software protection, protection of

physical components is also important. Any physical assault should be protected. Mechanisms used to ensure this requirement are: locked door, fixed components, security guards and component's access even after theft.

2.3.2 Additional Security Requirements

Along with the above-mentioned requirements, trust is identified as an additional security requirement in [38].

- **Trust:** Trust can be ensured in IoT systems in two ways. One way is to establish trust in the interaction between the endpoints, and also taking care of any mistrust that can arise in future collaborations among different endpoints. The second way is to maintain the trust of the user in the system. The user must be the controller of the system and must not feel that some external entity is controlling the system.
- **Data Freshness:** This requirement ensures that only the latest data is used. The use of latest data ensures that the need for real-time data is fulfilled.

2.4 Available Security techniques

This section will present the various techniques proposed in the literature for providing measures for the security issues raised in section 2.2. Based on the literature, the techniques are analyzed under the security services: confidentiality, access control, privacy, RFID security and secure routing. This analysis will be used while implementing security requirements of a particular system, i.e. smart home, that are also compatible with the constraints of IoT-based systems.

2.4.1 Confidentiality

Confidentiality means keeping an information secret. Confidentiality has to be provided for data storage and during data transmission. Confidentiality also ensures identification and authentication of an object. Authentication tells whether the authorized user is legitimate or not. If confidentiality and authentication are not taken care of, then there are chances of information leakage, data theft, spoofing, and masking. Cryptographic techniques for IoT have to be fast and should require fewer computations and memory. Confidentiality takes care of **Identification, Authentication and Privacy Security Requirements** as mentioned in section 2.3.1

Confidentiality is taken care of by cryptographic techniques like symmetric key algorithms such as AES, DES. Integrity is preserved by hash functions(MD, MD5, SHA-1, SHA-26).

Asymmetric key algorithms (RSA, ElGamal) require key management algorithms (Diffie-Hellman) for digital signatures and key transport over insecure channels. ECC (Elliptic curve cryptosystem) and cryptographic IC are seeking most of the attention.

Kakali et al. [17] proposed an authentication scheme based on Elliptic Curve Diffie–Hellman (ECDH). This scheme covers the issue of key management and access control in Wireless sensor networks (WSN) along with confidentiality. Authors have compared their proposed scheme with some previously proposed schemes based on ECDH. Their scheme outperforms the other ones regarding attack mitigation.

Thomas Kothmayr et al. [11] proposed a two-way authentication security scheme. The scheme includes Datagram Transport Layer Security (DTLS) handshake. It provides confidentiality and integrity.

Quangang Wen et al. in [12] introduced a dynamic variable cipher security certificate. This can be applied to IoT sensor layer.

In [19], IEEE 802.15.4 networks are secured through a standard compliance framework. This framework provides data confidentiality, integrity, lightweight solution and key management among nodes.

In [24], Kai Fan et al. proposed LRMAPC (lightweight RFID mutual authentication protocol with cache in the reader) for IoT. This lightweight protocol takes care of computational and transmission costs when a large number of tags are to be authenticated.

Wang Chen [13] proposed IBE (identity-based encryption). Along with fast and power efficient encryption, it provides authentication.

Siwei Peng[14] proposed an IMA (id based multiple authentication) technique. It prevents WSN from node replication and provides secure data aggregation and authentication.

Zhihua Li[15] proposed Public Key Infrastructure (PKI) like security mechanism to strengthen node authentication. A security foundation architecture and a protocol based on the research of PKI in TCP/IP are proposed.

Manju Suresh et al.[31] implements original and modified Blowfish on Xilinx Virtex-5 XC5VLX50T FPGA using Verilog HDL. Modified Blowfish algorithm takes 16.9% less execution time as compared to the original one. And the throughput increases by 18.7%.

Most of the proposed schemes are secure against man-in-the-middle attacks, denial of service attacks and provide perfect forward secrecy.

2.4.2 Access control

Access control is giving access to selective person, place or resource. Access control gives permission of authorized access. Access control tells who can access what. Violation of access control includes unauthorized access, security breach, intrusion and trespassing. Access control can be provided by locks and login credentials. Access control ensures **Authorization and Integrity Security Requirements**, which are described in section 2.3.1.

Attribute-Based Encryption (ABE) and Role Based Encryption (RBE) ensures access control. The proposed works are variant of these two. Sergio Gusmeroli et al. [35] claimed that authorization frameworks like RBAC (Role Based Access Control) and ABAC (Attribute Based Access Control) are not scalable, manageable, dynamic, effective, and efficient mechanisms to support distributed systems like IoT with many interacting services. A Capability Based Access Control (CapBAC) is proposed which supports rights delegation and more sophisticated access control techniques.

Nouha Oualha et al. [30] proposed an extended Ciphertext Policy-Attribute Based Encryption (CP-ABE). IoT devices like sensors and actuators because of their low computation cannot be used as enforcement points for CP-ABE. Therefore, a variant of CP-ABE is proposed.

2.4.3 Privacy

IoT is ubiquitous and interactive. These features can create opportunities for privacy violation. Privacy can hinder the development of IoT applications. Privacy can be voluntarily sacrificed, but involuntarily leakage of sensitive data and information is a violation of privacy. Data privacy and location privacy are the two main privacy issues in any IoT scenario. Physically based schemes, password based schemes, permissions, frameworks have been proposed for privacy protection [7][9]. Privacy ensures **Privacy Security Requirement**, described in section 2.3.1.

Shohreh Hosseinzadeh et al.[23] proposed an obfuscation/diversification of the operating system (OS) and an application interface (API) used in IoT. This will prevent an attacker from taking undue advantage of the system. But this obfuscation/diversification adds to the computational costs and memory utilization.

In [26], Mary R. Schurgot et al. applied the privacy preserving solutions to home automation test bed. The focus was on privacy preserving by using both cryptography and information manipulation.

Ikram Ullah et al.[28] enhanced the Semantic Obfuscation Technique (SOT) to preserve privacy in IoT. The proposed scheme i.e. ESOT provides location privacy.

Antonio J. Jara et al. [9] individually proposed security framework for the healthcare system in IoT environment. The security challenges, mainly privacy, are analyzed, and then solutions for them are proposed. These solutions were combined to design the security framework.

Attribute based signatures (ABS) is a possible solution for authentication with privacy. But, these existing techniques still have drawbacks regarding signer privacy or an expressive policy support. Jinshu Sua et al. [36] describes a signature scheme that uses an attribute tree and computational Diffie-Hellman. The comparison of the proposed technique with the existing ones shows that it outperforms them.

L. González-Manzano et al. [37] proposed an aggregation protocol for IoT settings and preserves privacy. The scheme uses Pallier cryptosystem. This scheme aims at providing privacy preserved aggregation protocol for IoT scenarios where a central sink node with multiple source nodes is present.

2.4.4 RFID system protocols

RFID raises different security concerns such as conflict collision, privacy, multiple tags, eavesdropping, tag authentication. RFID tracking and inventorying are the two main privacy concerns. RFID tags can be easily forged. Hence tag authentication is required. RFID is the basic unit of IoT. RFID security issues and requirements (like low power, low computation) need to be resolved.

RFIDs are the base of IoT systems. Following are some of the protocols used to provide security in a RFID system: Strong Private Authentication Protocol, Efficient Mutual-Authentication Protocol, Dimitriou's Lightweight Protocol and Advanced Semi-Randomized Access Control. EMAP does not support any strong encryption algorithm. It provides safety against replay attack and compromising resistance. Whereas, ASRAC prevents against cloning, Man in the middle attack, forward secrecy, tag anonymity, user data confidentiality, replay attack and compromising resistance[6].

In [24], Kai Fan et al. proposed a secure application revocation scheme in multi-application RFID which could improve the performance and security level of existing RFID scheme. It provides complete anonymity, confidentiality, anti-replay authentication

2.4.5 Secure Routing

Traditional communication techniques TCP, WAN, IPSec, are used at transportation layer. The security issues of these techniques also pose a threat to IoT security. The denial of service attacks can lead to unavailability of the system. The man-in-the-middle attack causes an information breach. Network paralysis attacks can halt the ongoing traffic. Hence, secure routing is needed. Secure routing ensures **Intrusion Detection and Immunity Security Requirements**, mentioned above in section 2.3.1.

Communication security prevents unauthorized intercepts from accessing telecommunications in an intelligible form, while still delivering content to the intended recipients. TLS/SSL protocols encrypt the link in the transport layer. Similarly, IPSec protects the network layer. Communication security protocols are designed to provide integrity, authenticity, and confidentiality in each layer.

M. Surendar et al. [29] proposed an InDRes (IDS for IoT with 6LoWPAN). This system detects sink hole attacks using constrained based specifications technique.

Somia Sahraoui et al. [20] proposed an end to end secure communication between the nodes and hosts is ensured by using compression models for HIP (Host Identification Protocol). Authors introduced a 6LoWPAN compression model for HIP (CD-HIP) and proved that compression and distribution models for HIP together are better than standard HIP.

In [16] Shahid Raza et al. designed and implemented a novel intrusion detection system for the IoT Environment.

In Table 2.2, the available work is divided into different headings. The table contains: Year (the year in which the work has been done), Author (who do the work), Proposed Technique (The techniques proposed/used in their work), Issues addressed (the security issues which are dealt with), Works at layer-P,T,A (it tells the layer where the mechanism implements security), Remarks and IoT Scenario (Security technique is proposed for which IoT application).

Table 2.2: Analysis of available security techniques

Year	Author	Proposed Technique/design	Issues addressed	Works at layer (P,T,A)	Remarks	IoT Scenario
2010 [9]	Antonio J. Jara et al	Propose an architecture to support IoT in medical environment	Secure communication, privacy, authentication	P,T,A	Prevents eavesdropping and denegation of service	Healthcare
2011 [10]	Liang Zhou, Han-Chieh Chao	Design a new and efficient multimedia traffic security framework	Key Management, Authentication, watermarking	P,T,A	-	Transportation
2012 [11]	Thomas Kothmayr et al	Proposed a 2-way authentication security scheme for IoT which includes DTLS handshake.	Confidentiality, integrity, availability	P,T,A	Efficiently uses energy, time and memory.	Works on top of standard communication stacks
2012 [12]	Quangan g Wen et al	Proposed application of dynamic variable cipher security certificate to IoT sensor layer.	Confidentiality, Information security, privacy	P	Reliable, robust	Remote car door opener
2012 [13]	Wang Chen	Propose IBE(Identity Based Encryption) based on ECC cryptosystem	Authentication, privacy	P	Solves man-in-the middle attack, information leakage, privacy, Cannot solve DoS, physical attacks	RFID, Sensor Networks
2012 [14]	Siwei Peng	Propose IMA(id based multiple authentication techniques)	Authentication, Secure data aggregation	P	Node replication attacks, DoS attacks	WSN
2013 [15]	Zhihua Li	Proposed PKI like security foundation architecture and protocol	Authentication	P	-	

2013 [16]	Shahid Raza et al.	Designed and implemented a novel intrusion detection system for the IoT	Intrusion detection, secure routing	P, T	Prevent routing attacks such as spoofing, sink hole and selective forwarding.	Contiki OS(OS for IoT)
2013 [17]	Kakali Chatterjee, Asok De, Daya Gupta	Proposed authentication scheme based on ECDH for WSNs.	Key management, access control, confidentiality, authentication	P	Prevents man-in-the middle attack, dictionary attack, stolen-verifier attack, node compromise attack, replay attack and provides perfect forward secrecy	WSN
2014 [18]	Teng Xu et al.	Initialization for creating CAD techniques that design highly optimized IoT devices	Physical safety, authentication	P	Hardware security primitive	-
2014 [19]	Savio Sciancalopore et al	Securing IEEE 802.15.4 networks through a standard compliance framework	Data confidentiality, integrity, lightweight solution, key management among nodes	T	-	OpenWSN stack
2014 [20]	Somia Sahraoui, Azeddine Bilami	Propose a 6LoWPAN compression model for HIP(CD-HIP)	End to end secure communication between nodes and hosts	T	-	-
2014 [21]	Kai Fan, Chen Liang et al	Proposed LRMAPC(lightweight RFID mutual authentication protocol with cache in the reader) for IoT	Tag authentication, reduce computational and transmission cost	P	Free from DoS, replaying, spoofing, eavesdropping, tracking	RFID tags

2014 [22]	Jun-Ya Lee, Wei-Cheng Lin et al	Proposed lightweight cryptography protocol	Authentication	P	-	RFID tags
2015 [23]	Shohreh Hosseinzadeh et al	Proposed obfuscation/diversification of the OS and API used in IoT	Information security, privacy	A	More memory consumption, execution overhead	IoT devices, sensors
2015 [24]	Kai Fan et al	Propose new RFID secure scheme	Complete anonymity, confidentiality, authentication	P	Replaying, recoverability	Multi-application RFID tag
2015 [25]	Ismail Butun, Burak Kantarci	To find potential solutions from anomaly detection aspect for security and privacy risks in IoT	Anomaly detection	-	-	Cloud-centric IoT
2015 [26]	Mary R. Schurgot et al	Apply privacy preserving solution based on cryptography and information manipulation to home automation test bed	Privacy	-	-	Home automation
2015 [27]	Ricardo Neisse et al	Proposed a Model based Security Toolkit, SecKit	Trust, privacy	P,T,A	Maintains trust ,trusted parties are allowed to do any changes	Smart home case study
2016 [28]	Ikram Ullah, Munam Ali Shah	ESOT, a model for preserving privacy in IoT-based on obfuscated location	Location privacy	-	Tracking	Android system for mobile devices
2016 [29]	M. Surendar , A. Umamak eswari	Propose InDRes(IDS for IoT with 6LoWPAN)	Secure routing, Integrity	T	Detect sink hole attacks using constraint-based specifications technique.	WSN(NS2 simulation)

2016 [30]	Nouha Oualha, Kim Thuat Nguyen	Proposed extended CP-ABE scheme for IoT	Access control, confidentiality, lightweight solution	P,A	Mitigate security issues in resource constrained devices	Sensors
2016 [31]	Manju Suresh, Neema M.	Implemented modified Blowfish algorithm on hardware.	Confidentiality, authentication	P	Less execution time, more throughput	FPGA
2016 [32]	S.R. Moosavi et al.	Propose an end-to-end security scheme for mobility enabled healthcare Internet of Things (IoT) based on DTLS handshake, session resumption, and interconnected smart gateways.	Confidentiality, authentication, access control, end to end security	-	Lightweight solution free from DoS attacks, sensor spoofing	Healthcare
2016 [33]	Fagen Li et al	proposed a heterogeneous encryption scheme to control the access behavior of the users in WSN	Authentication, access control	P	Reduced computational cost and energy consumption	WSN
2016 [34]	Vu Mai, Ibrahim Khalil	proposed a cloud-based data storage and processing model	Privacy, confidentiality	A	Homomorphic computing model is used	Smart grid(Smart meter)

2.5 Security Engineering

"Security engineering is a specialized field of engineering that focuses on the security aspects in the design of systems that need to be able to deal robustly with possible sources of disruption, ranging from natural disasters to malicious acts."[46]

Security engineering framework is an attempt to structure the complex task of providing security to any system. Security engineering framework should work in parallel with the software development life cycles. Both the system specification and system security requirements should be identified in the Requirements Engineering phase. Similarly, design, implementation, and testing should go side by side. Security engineering ensures that security requirements are identified along with the systems' functional and non-functional requirements.

Kakali et al.[40] are the first ones to propose a security engineering framework. Figure 2.2 depicts the framework proposed by them, which is divided into four phases.

1) Security Requirements Engineering

In this phase, stakeholders for the system are identified. Security requirements are identified along with the functional and non-functional requirements. Threats and assets are identified. Based on the risk assessment result, security requirements are then prioritized.

2) Security Design Engineering

In the design phase, security requirements are mapped to the security mechanisms. Based on the environment and device constraints, efficient algorithms are chosen from the repository. This prevents maximum attacks, thus achieving the security requirements and goals.

3) Security Implementation

During the implementation, the selected algorithms are implemented.

4) Security Testing

In the testing phase, the attackers and the attacks are identified. The system is then tested for the implemented security requirements against the identified attacks possible on the system.

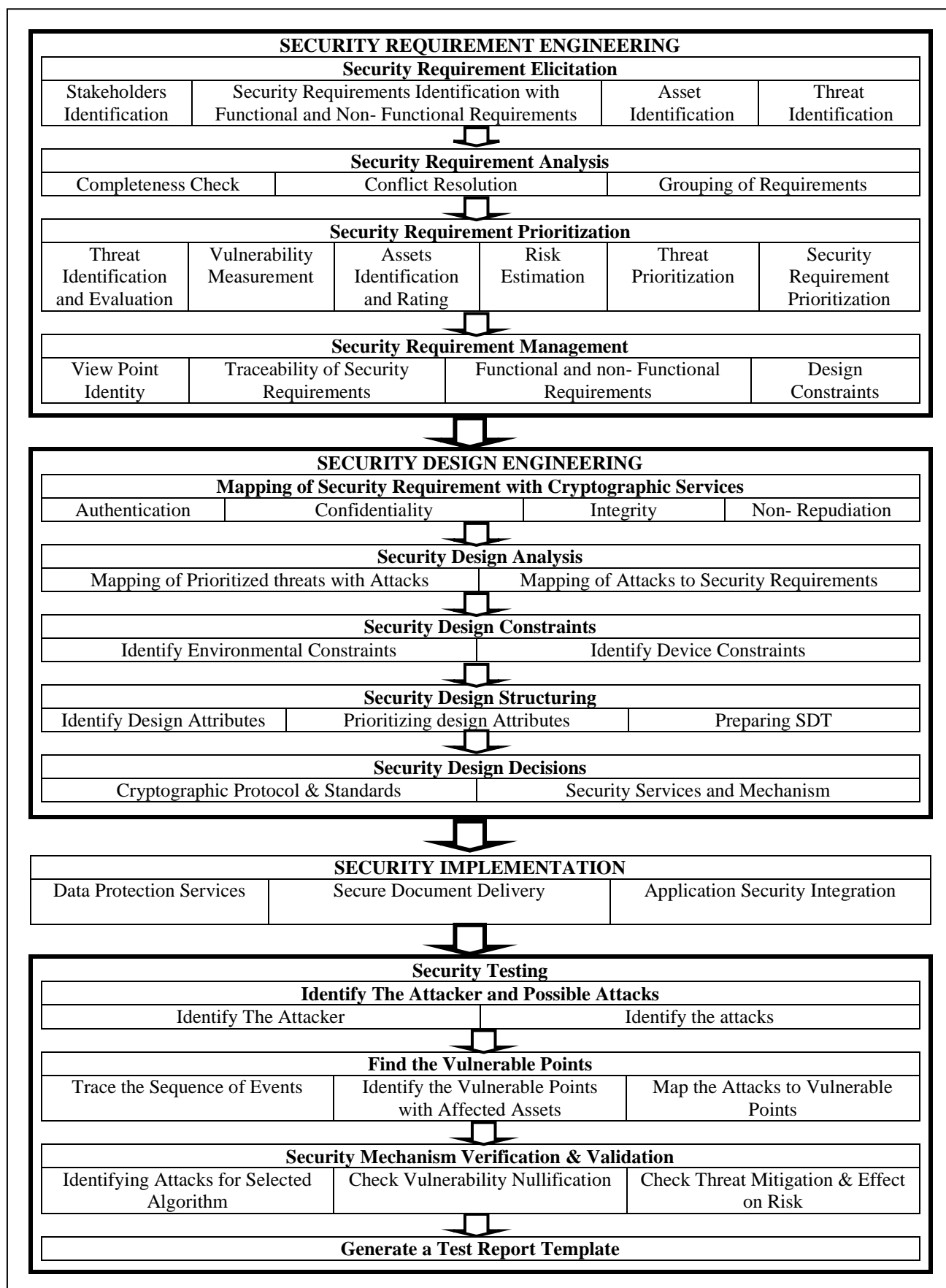


Figure 2.2: Security Engineering Framework[40]

Security Requirements Engineering for IoT

This chapter discusses the modified security engineering framework. In the later parts, it explains the security requirements engineering phase for generic "home automation systems", an IoT application. The security design engineering phase is described in the next chapter.

3.1 Modified Security Engineering Framework for IoT

As described in section 2.5, the security engineering framework[40] is proposed for generic systems, which should be followed along with the software development life cycle. This framework ensures that all the threats and security requirements for the system are identified from the very beginning of the project.

Here, we are focusing only on the first two phases of the generic security engineering framework presented in the previous section. Modified security engineering framework is explained in Figure 3.1.

Phase I: Security Requirements Engineering- This phase consists of four steps. During requirements elicitation, the assets, actors, threats, vulnerabilities, and security requirements are identified. During analysis, the interdependencies among assets and actors, actors and vulnerabilities, assets and vulnerabilities, assets and threats, threats and vulnerabilities are identified. Finally, the risk is calculated for each threat using fuzzy logic. Based on the risk values, threats are prioritized as high, medium and low risk threats.

Phase II: Security Design Engineering- This phase consists of three steps. The security requirements are associated with the security services for a system. Pre-existing security algorithms are identified, and a repository of all such algorithms is created. On the basis of threats mitigated, a security index is calculated for these algorithms. The criteria for selecting algorithms for implementation is security index, rather than ad hoc basis.

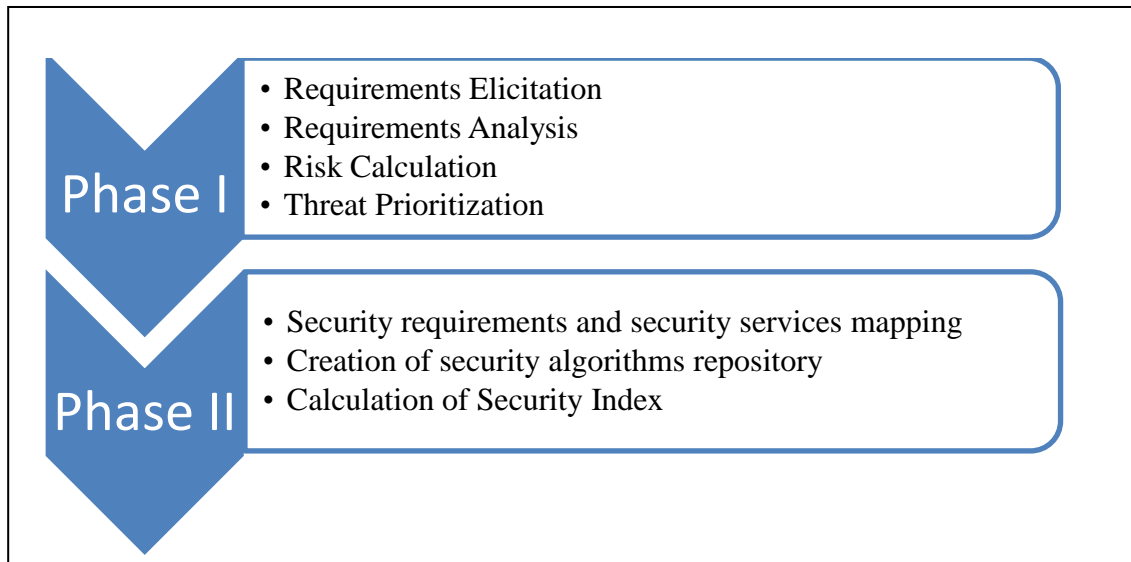


Figure 3.1: Modified Security Engineering Framework for IoT

3.2 Security Requirements Engineering

In this phase, the security requirements of the system are identified. For the identification of these requirements; actors, assets, vulnerabilities, and threats of the system are identified. Threats are then prioritized on the basis of risk values. This phase is carried out in the following steps:

- 1) Requirements Elicitation
- 2) Requirements Analysis
- 3) Risk Calculation
- 4) Threat Prioritization

3.2.1 Requirements Elicitation

Requirements Elicitation means gathering requirements of the system from various sources. Requirements are gathered from various sources, both offline and online. Different IoT-based systems and applications are studied. On the basis of this literature survey[41][42], the requirements for the smart home system are identified. For the identification of the actors of the system, requirement engineering phase of a software development life cycle is studied. Figure 3.2 shows the steps taken in Requirements Elicitation phase.

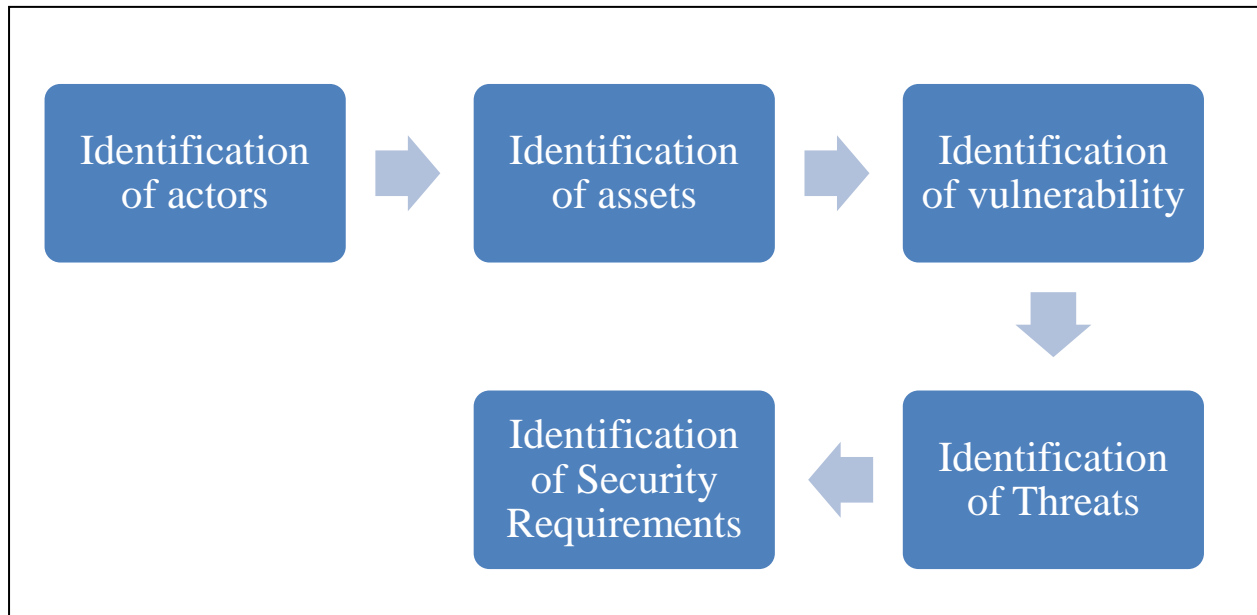


Figure 3.2 Steps in Requirements Elicitation

➤ **Identification of Actors**

An actor models a role played by an entity (external or internal) or specifies a user or a system component which interacts with the system. "Role" refers to the behavior of the user with respect to the situation. Viewpoint-oriented approach [41][42] is used to identify the different actors or stakeholders. Actors are identified as direct or indirect actors. Direct actors are those who interact directly with the system and have liability. Indirect actors play some role in the functionalities of the system but have no liability. Based on this approach, following actors are identified for home automation systems:

- **Users**

The user is an entity for whom the system is designed. The user has access to the functionalities of the system and use the system or play a role in the system. In our case study, for a home based automation system, the user can be the owner, inhabitant of the home, friend of the inhabitants or guest.

- **Communication Channel**

Communication channel links the remote users to the system. Internet network is used as a communicating medium.

- **Interfaces**

Interfaces are the devices which have the software of the system installed on them. Using the interface devices, the user can use, monitor and control the system. Laptops, PDA, tablets or smartphones act as interfaces in our IoT-based system.

- **Service Providers**

Service providers are the industries or the companies who design, create, distribute and maintain the system. Service providers are responsible for maintenance of devices and internet connectivity.

- **End Point applications**

End point applications are the software applications which run on the interfaces. Website and a smart phone application are the examples of end point applications and run on a web browser enabled laptop and smart phone respectively.

- **Gateway**

The system which makes the communication between the devices and the interfaces possible by connecting them to the internet.

- **Devices**

Devices can be end-point devices (like sensors, electronic appliances), peer devices (which communicate with the system indirectly via others) and intermediate devices (sensor controllers, etc. which connects the peer devices to the gateway and monitors them).

➤ **Identification of Assets**

Assets are the valuable resources and the liabilities of a system. Assets for a system can be any physical entity or data. Generic assets for home automation system are identified on the basis of the literature survey [41][42]. Each asset belongs to different actors. That is why, assets are mapped to actors in Table 3.1.

Table 3.1 : Assets corresponding to actors

Assets	Actors involved
System Data	User Communication Channel Interfaces Endpoint applications Service providers Gateway Devices
Privacy	User Communication Channel
Logs	Endpoint applications Service providers Gateway Devices
Network	Communication Channel Service providers Gateway
Efficient working of the system	User Communication Channel Interfaces Endpoint applications Service providers Gateway Devices
Trust	User Communication Channel Interfaces Endpoint applications Service providers Gateway Devices
Devices	User Endpoint applications Devices
Data storage units	Users Service providers

➤ Identification of Vulnerabilities

The vulnerability is referred as a weakness of the system due to which the security of the system can be compromised. Attackers exploit the flaws present in the functionalities of the system and actors are the ones who use the system's functionalities. Therefore, vulnerabilities are mapped to the actors identified in section 3.2 on the basis of the functionalities accessed by the actor. Various vulnerabilities are identified for our case study on the basis of literature survey[41][42]. Table 4.2 shows the vulnerabilities identified for each actor.

Table 3.2 : Identification of vulnerabilities based on actors

Actors	Vulnerability
Users	Naive user Weak access control
Communication Media	Unencrypted data Insecure network Obsolete systems Interoperability issues
Interfaces	Security breach Unauthorized access Information leakage Misconfiguration System failure Physical security Lack of standards Monitoring absence Old data Logging
End points	Physical Security No encryption Remote Access Malware attack Information leakage Intrusion Obsolete data Inaccurate data Device failure Insufficient security configuration Eavesdropping
Service providers	Insecure interfaces Weak firewalls Insecure network services

Gateways	Insecure communication Unauthorized access Inefficient Firewalls Insecure interfaces Improper logging Intrusion Certification Insufficient security configuration Weak cryptographic techniques Malware attacks System misuse
Intermediate Devices	Insecure system configuration

➤ **Identification of Threats**

Threats are the reason for security breach. Exploiting a vulnerability, a threat causes harm or damage to the system. Potential threats for the system are identified from the literature survey [41][42]. Table 3.3 depicts the identified threats which are mapped to the vulnerabilities of the system. A mark (X) in the table 3.3 implies that the corresponding vulnerability leads to that threat.

Table 3.3 : Threat-Vulnerability mapping

Threat→ Vulnerability↓	T. Fraud	T. Data_theft	T. identity_theft	T. credential_theft	T. spoofing	T. phishing	T. DDoS	T. Malware_attack	T. security_breach	T. unavailability	T. system_failure	T. hardware_failure	T. software_crash	T. eavesdropping	T. privacy_violation	T. system_misuse	T. technical_failure	T. power_failure	T. flooding_attack	T. node_attack	T. node_capture	T. change_data	T. man_in_the_middle_attack	T. communication_change	T. Data_leakage
Naive user						X							X									X			
Weak Access Control	X		X	X	X				X						X							X			
Unencrypted data		X												X								X	X	X	X
Insecure networks		X				X	X		X	X						X	X		X			X	X	X	
Obsolete systems										X	X						X					X			
Interoperability Issues										X	X	X	X				X								

identification security requirement mitigates T.Identity_theft, T.Spoofing, and T.Fraud.

The security requirements identified for the home automation system are:

- 1. Identification Security Requirements:** To fulfill this requirement, each user and component of the system should be known to the system. If a visitor comes, then he should provide all the identity details.
- 2. Authentication Security Requirements:** To fulfill this requirement, every user and component of the system are to be authenticated first and then provided with the permission to use the system.
- 3. Authorization Security Requirements:** To fulfill this requirement, the system checks whether the user is authorized to do the certain task in the system or not. It should allow only the authorized users to do certain tasks.
- 4. Privacy Security Requirements:** To fulfill this requirement, the system should limit the sharing permissions. Only essential and non-personal data should be shared. And if personal data is shared, then it should be encrypted and should be shared with the trusted and authorized parties only.
- 5. Physical Protection Security Requirements:** To fulfill this requirement, the system should safeguard it against theft, destruction or unidentified replacement. The main door should always be kept locked and only authorized, and known visitors should be allowed to enter the home.
- 6. Integrity Security Requirements:** To fulfill this requirement, the system should ensure that message once sent is not modified later. Digital certificates are used to ensure this requirement. Also, the message should be encrypted well before sending.
- 7. Immunity Security Requirements:** To fulfill this requirement, the system should prevent any unauthorized access. The system settings should be changed only by the administrator. It should detect the presence of malware or trojans in the system and find out ways to eradicate them.
- 8. Intrusion Detection Security Requirements:** To fulfill this requirement, the system should recognize any abnormal activity as soon as it happens in the system. Alarms should be there to inform any intrusion.
- 9. Security Maintenance Security Requirements:** To fulfill this requirement, the status of the implemented security algorithms is to be checked regularly. The software

should be updated regularly so that the system is compatible with the ongoing advanced technologies.

10. Survivability Security Requirements: To fulfill this requirement, the system should recover quickly from any adverse situation like power failure, device failure, intrusion detection, security attack or natural disaster. Some minimal functionalities should be set which will keep the system going even under these situations.

11. Trust Security Requirements: To fulfill this requirement, the system should distinguish between the trusted and non-trusted components.

Table 3.4 : Security requirements-Threats Mapping

Security Requirements→ Threats↓	Identification	Authentication	Authorization	Immunity	Integrity	Intrusion detection	Privacy	System Maintenance	Survivability	Physical Protection	Trust
T. Fraud	X	X			X						X
T. Data_theft		X	X								X
T. identity_theft	X	X	X								X
T. credential_theft		X	X								X
T. spoofing	X	X	X								X
T. phishing		X	X								X
T. DDoS		X		X					X		X
T. Malware_attack				X					X		X
T. security_breach			X			X					X
T. unavailability										X	X
T. system_failure								X			X
T. hardware_failure								X			X
T. software_crash								X			X
T. eavesdropping							X				X
T. privacy_violation			X				X				X
T. system_misuse			X		X						X
T. technical_failure								X			
T. power_failure								X			
T. flooding_attack				X							X
T. node_attack						X					X

T.node_capture			X			X				X	X
T.change_data		X	X								X
T.man_in_the_middle_attack						X	X				X
T.communication_change									X		X
T.Data_leakage											X
T.Routing_attack				X							

3.2.2 Requirements Analysis

During requirements analysis, the parameters which define the impact of the requirements and their mappings are calculated. Requirements analysis is performed to make sure that all the possible combinations of the requirements identified in the previous section are considered so that the calculated parameters cover all aspects of security. This analysis is done to calculate risk and prioritize the threats on the basis of risk. The different mappings done in this section are inspired from OWASP guidelines. The different mappings done in this step are:

- **Asset and Actor mapping (Asset Rating)**

Assets can belong to one or more actors. For example, user has personal data, privacy, efficient working of the system, trust, devices and data storage as his assets. Table 3.5 shows which asset belongs to which user via a matrix. Asset rating is calculated as the sum of all the actors for an asset.

Table 3.5: Asset and actor mapping (Asset Rating)

Assets↓	Actors→							Asset rating
	Users	Communication channel	Interfaces	Endpoint applications	Service providers	Gateway	Devices	
Personal Data	X	X	X	X	X	X	X	7
Privacy	X	X						2
Logs				X	X	X	X	4
Network		X			X	X		3
Efficient working of system	X	X	X	X	X	X	X	7
Trust	X	X	X	X	X	X	X	7
Devices	X			X			X	3
Data storage	X				X			2

- **Vulnerability and Actor mapping**

Vulnerability and actor mapping is done to get the rating for each vulnerability. This vulnerability rating gives information about the number of actors a vulnerability effects and will then be used as weights in the threat and vulnerability mapping (Table 3.8). Table 3.6 shows the vulnerability and actor mapping in a matrix form. Vulnerability rating is calculated as the sum of marks in a particular row.

Table 3.6: Vulnerability and actor mapping (Vulnerability Rating)

Vulnerability↓	Actors →							Vulnerability Rating
	Users	Communication channel	Interfaces	Endpoint applications	Service providers	Gateway	Devices	
Naive	X							1
Weak Access Control	X			X		X	X	4
Unencrypted data		X	X	X			X	4
Insecure networks		X						1
Obsolete systems		X	X				X	3
Interoperability Issues		X			X			2
Physical security			X	X			X	3
Malware attack				X			X	2
Information leakage			X	X			X	3
Intrusion detection				X			X	2
Obsolete data				X			X	2
Inaccurate data			X	X				2
Device/system failure			X	X				2
Inefficient security configurations				X			X	2
Eavesdropping/ resource isolation				X			X	2
Insecure interfaces				X	X			2
Insecure network services					X		X	2
Improper/inefficient logging						X	X	1
Misconfiguration			X				X	2
Lack of standards			X		X		X	3

Firewall inefficiency					X			1
Monitoring absence			X					1
System's resources misuse						X	X	2
Weak cryptographic techniques			X					1
Unauthorized access			X	X		X		3

- Threat and Asset mapping (Impact Rating)**

A threat can cause harm to the system. Assets and actors are the ones effected by this harm. The assets are the targets in any attack. Therefore, to know the effect of any threat, the assets associated with any threat are to be considered. Table 3.7 shows the assets corresponding to each threat. These assets are rated above based on the actors and asset mapping. Impact rating of the threat is calculated as sum of the asset ratings. This impact rating specifies the extent to which a threat is harmful to the system.

Table 3.7: Threat - Asset Mapping (Impact rating)

Threat↓	Asset→								Impact Rating
	Personal Data	Privacy	Logs	Network	Efficient working of system	Trust	Devices	Data storage	
T. Fraud						7			7
T. Data_theft	7	2				7		2	18
T. identity_theft	7					7			14
T. credential_theft		2				7			9
T.spoofing	7			3		7		2	19
T. phishing						7			7
T. DDoS				3					3
T.Malware_attack							3		3
T.security_breach	7	2				7		2	18
T.unavailability					7				7
T.system_failure			4	3	7		3		17
T.hardware_failure					7		3		10
T.software_crash					7		3		10
T.eavesdropping		2				7			9
T.privacy_violation	7	2							9
T.system_misuse						7			7
T.technical_failure					7				7
T.power_failure					7				7

T.flooding_attack				3	7				10
T.node_attack							3		3
T.node_capture						7	3		10
T.change_data	7					7			14
T.man_in_the_middle_attack		2				7			9
T.communication_change				3		7			10
T.Data_leakage	7	2				7			16
T.Routing_attack				3		7			10

- Threat and Vulnerability mapping (Threat Rating)**

Threat and vulnerability mapping tell about the vulnerabilities a threat can exploit. Based on this number of vulnerabilities the threat rating is calculated. Table 3.8 maps the threats and vulnerability. The vulnerability weights are calculated by vulnerability-actor mapping. Threat rating is calculated as the sum of vulnerability ratings for each threat. Threat rating tells the probable chances of occurring of any threat.

Table 3.8: Threat and Vulnerability (Threat Rating)

Threat→ Vulnerability↓	T. Fraud	T. Data_theft	T. identity_theft	T. credential_theft	T.spoofing	T. phishing	T. DDoS	T.Malware_attack	T.security_breach	T.unavailability	T.system_failure	T.hardware_failure	T.software_crash	T.eavesdropping	T.privacy_violation	T.system_misuse	T.technical_failure	T.power_failure	T.flooding_attack	T.node_attack	T.node_capture	T.change_data	T.man_in_the_middle_attack	T.communication_change	T.Data Leakage
	Naive user						1							1									1		
Weak Access Control	4		4	4	4				4						4							4			
Unencrypted data		4												4								4	4	4	4
Insecure networks		1				1	1		1	1						1	1		1			1	1	1	
Obsolete systems										3	3						3					3			
Interoperability Issues										2	2	2	2				2								
Physical security	3							3	3					3	3	3				3	3	3			
Remote access																									
Malware attack					2		2	2	2			2	2			2	2			2	2				
Information leakage		3	3	3										3	3	3									3
Intrusion	2				2			2	2					2	2	2									
Obsolete data						2																2			
Inaccurate data	2				2	2																2			
Device/system								2	2	2	2	2	2				2	2				2			

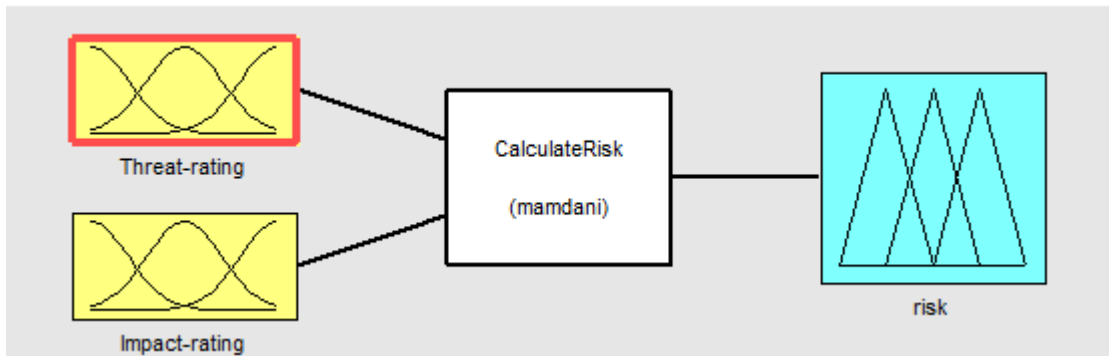


Figure 3.3 : Fuzzy system for calculating risk

Following steps are taken for calculating the risk using fuzzy logic:

1) Membership Function for Input 1: Threat Rating

Threat rating is calculated in Table 3.8, by counting the total of vulnerability ratings for each threat. The maximum value of a threat can be 50 and minimum is 0 shown in Table 3.8. Therefore, the membership function lies between [0, 50]. Table 3.9 gives the numeric value range of the linguistic variables (VL, L, M, H, VH) for threat rating and Figure 3.4 depicts the membership function plots of the threat rating.

Table 3.9: Weights of Threat Rating Variable

Threat Rating Value	Variable
0-5-10	Very Low (VL)
5-10-20	Low (L)
10-20-30	Medium (M)
20-30-40	High (H)
30-40-50	Very High (VH)

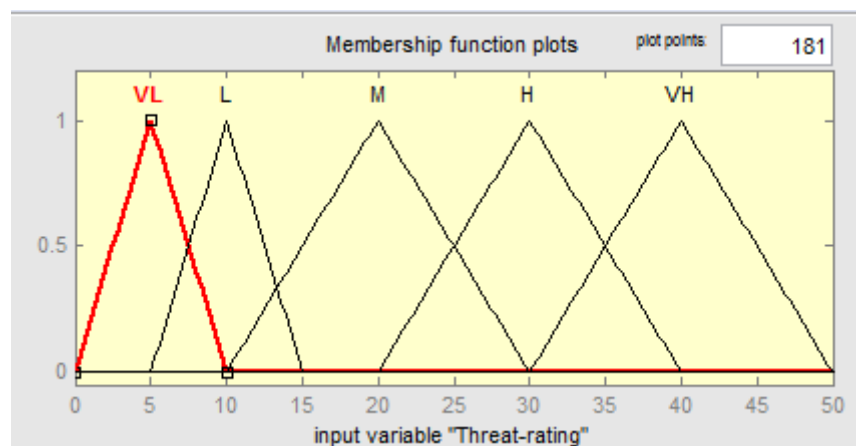


Figure 3.4: Membership Function: Threat Rating

2) Membership function for Input 2: Impact Rating

Impact rating is calculated in Table 3.7, by counting the sum total of asset ratings for each threat. The maximum value of a threat can be 20 and minimum is 2 shown in Table 3.7. Therefore, the membership function lies between [2, 20]. Table 3.10 gives the numeric value range of the linguistic variables (L, M, H) for impact rating and Figure 3.5 depicts the membership function plots of the impact rating.

Table 3.10: Weights of Impact Rating Variable

Impact Rating Value	Variable
2-7-7	Low (L)
7-10-15	Medium (M)
10-15-20	High (H)

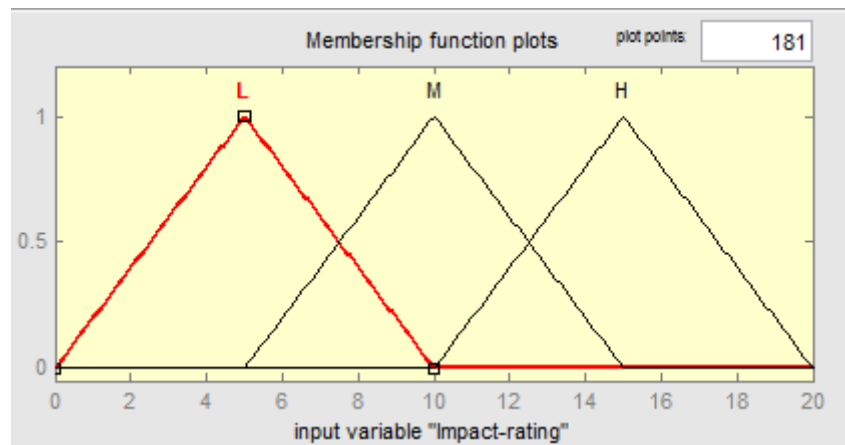


Figure 3.5: Membership Function: Impact Rating

3) Membership Function for Output: Risk

The risk is calculated on the basis of threat rating and impact rating. Based on both the inputs, the maximum value of risk can be 1000. Therefore, the membership function lies between [0, 1000]. Table 3.11 gives the numeric value range of the linguistic variables (L, M, H) for Risk and Figure 3.6 depicts the membership function plots of Risk.

Table 3.11: Weights of Risk variable

Risk Value	Variable
0-0-300	Low (L)
100-400-700	Medium (M)
500-750-1000	High (H)

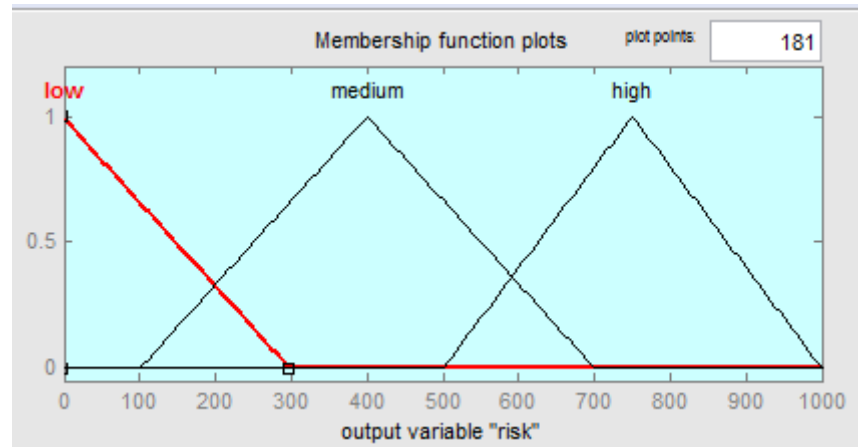


Figure 3.6: Membership Function: Risk

4) Fuzzy Rules

Fuzzy rules are like If-then statements, for example: If it is raining, then take the umbrella. Fuzzy rules give fuzzy the human-like intuition. A fuzzy rule is stated as "If input1=variable and input2=variable, then output=variable." For this system, we have five variables (VL, L,M,H,VH) for threat rating and three variables (L, M, H) for impact rating. Hence, the fuzzy inference system will have 15 rules. Some of these rules are mentioned in the Table 3.12. These rules are designed on the basis of understanding of the system.

Table 3.12: Rules for the Fuzzy system

S. No.	Rule	Output
1	If (Threat-rating is VL) and (Impact-rating is L)	L
2	If (Threat-rating is L) and (Impact-rating is M)	M
3	If (Threat-rating is M) and (Impact-rating is M)	M
4	If (Threat-rating is H) and (Impact-rating is H)	H
5	If (Threat-rating is VH) and (Impact-rating is H)	H

5) Risk Calculation

In fuzzy systems, after setting the membership values and rules, the output is calculated through defuzzification. Following is an example showing, how risk value is calculated for given values of threat-rating and impact-rating. Figure 3.7 depicts the calculation of the risk for given value of threat rating (25) and impact rating (10). The fuzzy system applies the rules based on the input values. The final value of the output is calculated on the basis of the value of the overlapping areas of the output membership function. In this work, the output

value is calculated as "center of gravity" of the overlapping areas. Table 3.13 depicts the risk value of the threats.

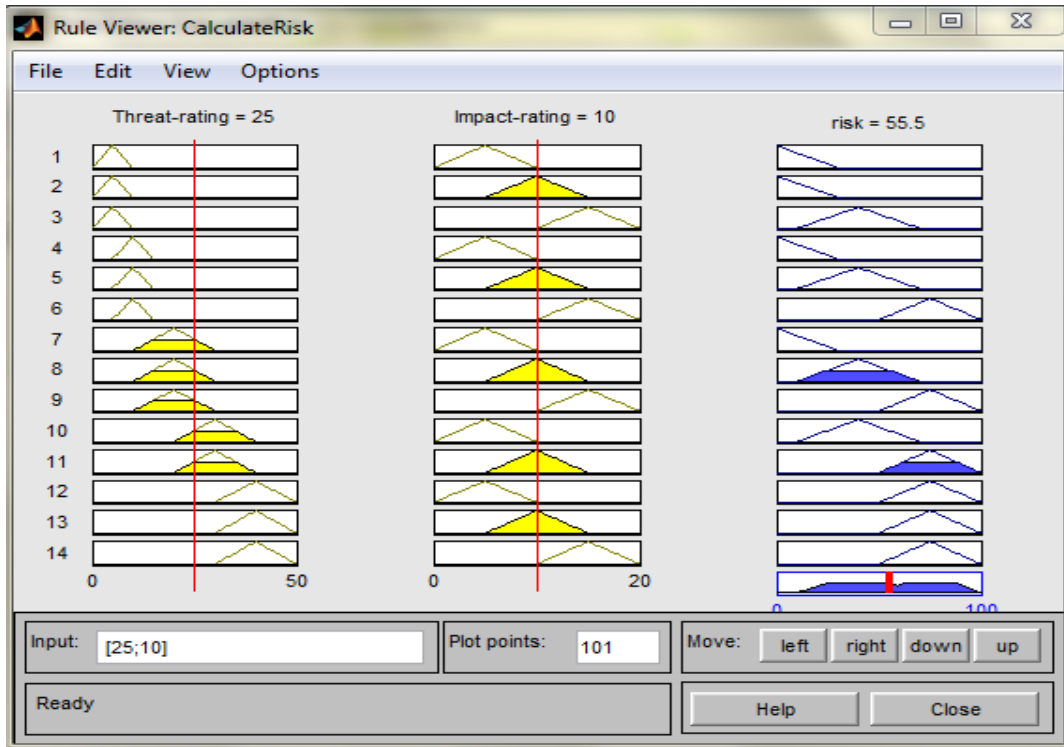


Figure 3.7: Defuzzification: Finding value of Risk for input [25;10]

Table 3.13: Calculation of Risk for threats

Threats↓	Threat Rating	Impact Rating	Fuzzy risk
T. Fraud	15	7	30.402
T. Data_theft	14	18	75
T. Identity_theft	11	14	64.534
T. Credential_theft	11	9	37.1164
T.Spoofing	13	19	75
T.Phishing	8	7	29.2572
T.DDoS	11	3	10.8760
T.Malware_attack	16	3	10.8760
T.Security_breach	21	18	75
T.Unavailability	13	7	31.9039
T.System_failure	9	17	63.4475
T.Hardware_failure	6	10	23.0117
T.Software_crash	9	10	37.1164
T.Eavesdropping	15	9	36.3955
T.Privacy_violation	20	9	37.1164
T.System_misuse	16	7	29.2572

T.Technical_failure	11	7	29.2572
T.Power_failure	2	7	11.9932
T.Flooding_attack	5	10	9.667
T.Node_attack	5	3	10.8760
T.Node_capture	5	10	9.6667
T.Change_data	34	14	75
T.Man_in_the_middle_attack	11	9	37.1164
T.Communication_change	12	10	40
T.Data_leakage	16	16	75

3.2.4 Threat Prioritization

Based on the risk values, threats are prioritized as high, medium and low-risk threats. Higher the risk value, higher will be the priority. Table 3.14 depicts the threats prioritized. Threats are prioritized as: Low risk (Risk value<35), medium risk (35<= Risk value <60) , and high risk (60<= Risk Value<100).

Table 3.14: Prioritization of threats

Low risk	Medium risk	High risk
T.Fraud	T.Communication_change	T.Data_leakage
T.Phishing	T.Credential_theft	T.Change_data
T.DDoS	T.Software_crash	T.System_failure
T.Malware_attack	T.Eavesdropping	T.Spoofing
T.Unavailability	T.Privacy_violation	T.Identity_theft
T.Hardware_failure	T.Man_in_the_middle_attack	T.Data_theft
T.System_misuse		T.Security_breach
T.Technical_failure		
T.Power_failure		
T.Flooding_attack		
T.Node_attack		
T.Node_capture		

Security Design Engineering for IoT

This chapter discusses the design phase of the framework. In this chapter, the security requirements are mapped to the security services. Security algorithms pool is created based on the services provided. Security index is calculated for the chosen algorithms.

4.1 Security Design Engineering

In this phase, the identified security requirements are mapped to the threats and a combined risk value is calculated for each security requirement. These security requirements are then mapped to the security services. A repository of available security algorithms is created. Security index is calculated for each algorithm. This phase provides mechanism for selection of efficient algorithms. Figure 4.1 depicts the steps followed during the design phase.

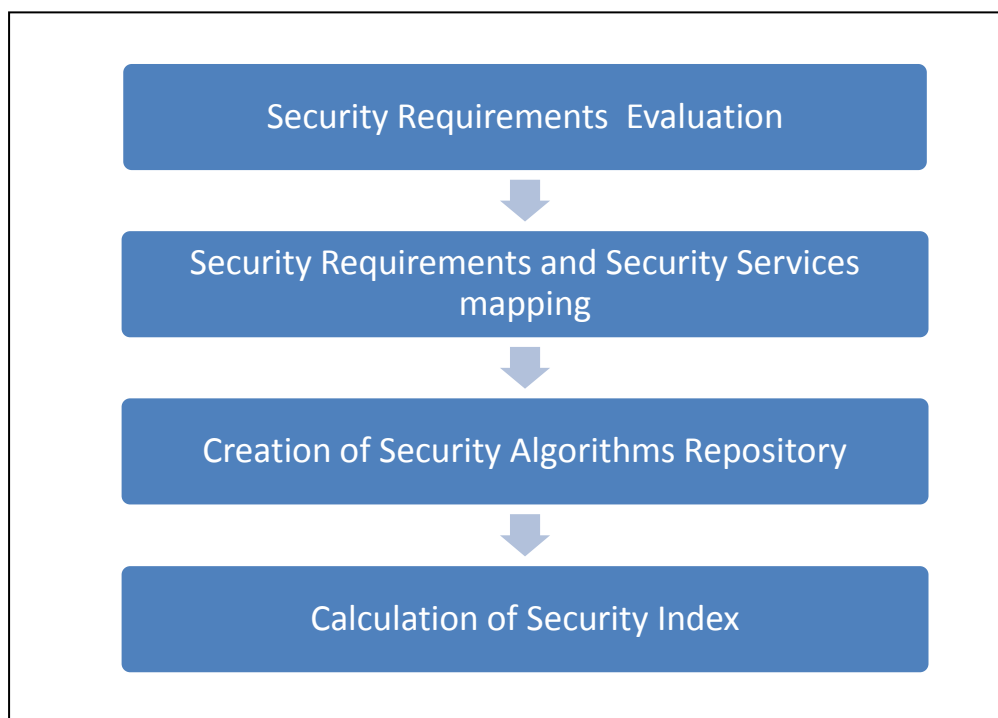


Figure 4.1: Steps of Security Design Engineering Phase

4.1.1 Security Requirements Evaluation

Risk value for each security requirements is calculated as a triplet (H,M,L), where H, M, L is the number of high risk, medium risk and low risk threats corresponding to the security requirement respectively. Table 4.1 depicts the risk values of the security requirements.

Table 4.1: Risk Values corresponding to Security Requirements

Security Requirements	Threats	Rating
Identification (H,M,L)	T.Identity_theft T.Spoofing T.Fraud	H H L (2,0,1)
Authentication (H,M,L)	T.Data_theft T.Identity_theft T.Spoofing T.Change_data T.Credential_theft T.Phishing T.Fraud T.DDoS	H H H H M L L L (4,1,3)
Authorization (H,M,L)	T.Data_theft T.Identity_theft T.Spoofing T.Change_data T.Security_breach T.Credential_theft T.Privacy_violation T.Phishing T.Node_capture T.System_misuse	H H H H H M M L L L (5,2,3)
Immunity (H,M,L)	T.DDoS T.Malware_attack T.Flooding_attack T.Routing_attack	L L L L (0,0,4)
Integrity (H,M,L)	T.Fraud T.System_misuse	L L (0,0,2)

Intrusion detection (H,M,L)	T.Security_breach T.Man_in_the_middle_attack T.Node_attack T.Node_capture	H M L L (1,1,2)
Privacy (H,M,L)	T.Man_in_the_middle_attack T.Privacy_violation T.Eavesdropping	M M M (0,3,0)
System Maintenance (H,M,L)	T.System_failure T.Software_crash T.Hardware_failure T.Technical_failure T.Power_failure	H M L L L (1,1,3)
Survivability (H,M,L)	T.Malware_attack T.DDoS T.Communication_change	L L M (0,1,2)
Physical protection (H,M,L)	T.Node_capture T.Unavailability	L L (0,0,2)
Trust (H,M,L)	T. Data_theft T. identity_theft T. credential_theft T.spoofing T.system_failure T.security_breach T.change_data T.software_crash T.eavesdropping T.privacy_violation T.man_in_the_middle_attack T.communication_change T. phishing T. DDoS T.Malware_attack T. Fraud T.unavailability T.hardware_failure T.system_misuse T.flooding_attack T.node_attack T.node_capture	H H H H H H H M M M M M M L L L L L L L L L L (7,5,10)

4.1.2 Security Requirements and Security Services mapping

Security algorithms provide security services. Confidentiality, Integrity and Availability (CIA) are the well known security services [44]. Trust is also added. These services provide the security requirements.

Table 4.2 depicts the security services and their corresponding security requirements and their security mechanisms..

Table 4.2: Security Requirements and Security Services mapping

Security Services	Security Requirements	Security Mechanisms
Confidentiality	Privacy Immunity	Encryption mechanisms
Integrity	Integrity	Hashing
Availability	Identification Authentication Authorization Survivability Intrusion detection Physical Security System Maintenance	Digital Certificates Authentication Exchanges Key agreement protocols Access control schemes Recovery services Intrusion detection scheme Maintenance services
Trust	Trust	Trust is ensured if all other security requirements are fulfilled.

4.1.3 Creation of security algorithms repository

For providing security, cryptographic algorithms are used. Previous section shows that cryptographic techniques deals with majority of security requirements. In IoT-based systems, Asymmetric, symmetric, signature, hashing and hybrid cryptographic algorithms are used. A repository consisting of popular algorithms of each type is created. Table 4.3 depicts the available and in use security algorithms.

Table 4.3: Security algorithms pool

Type of Algorithms	Examples
Symmetric Algorithms	AES DES Triple DES
Asymmetric Algorithms	RSA ECC HECC
Hashing Algorithms	MD4 MD5 SHA1
Signature Algorithms	RSA + DSA ECDSA HECDSA
Hybrid Algorithms	N/2 (AES + ECC) + N/2 (DUAL RSA) + HASH ECC + DUAL RSA + MD5 Lightweight Hybrid Cryptographic Algorithm ECIES

Design Constraints of IoT

IoT security is different from Network security [38]. In IoT, devices have less memory and less computational power, therefore system cannot run complex and high power security protocols. Also, the devices have low power, this energy constraint complicates the search for security solutions.

Since, the number of IoT devices is increasing rapidly, there arises another issues such as network jamming (because of the increase in the traffic), addressing issues (a rise in number of devices will lead to identification issues), collisions and confusion among the short wavelength transmissions, power and storage constraints.

These characteristics limit the solutions for security. The security solutions hence should:

- be light weight
- require less computation
- consume less power
- be embedded (RFID's)
- need less memory

4.1.4 Calculation of Security Index

Security index is calculated as the ratio of threats mitigated and the possible threats of the system. Security index value ranges from 0 to 1. Higher the security index, better the

algorithm is. Security index is the measure of the efficiency of a security algorithm. Based on security index value algorithms are selected for the implementation phase.

From the pool of security mechanisms as shown in Table 4.3, some mechanisms are selected and are mapped against the security requirements. Based on the requirements fulfilled by these mechanisms, security index is calculated for each mechanism. Table 4.4 depicts mapping between security requirements and security mechanisms. This table is then used to calculate the security index.

Based on the requirements fulfilled, the (H, M, L) value is calculated as the sum of all corresponding requirements' (H,M,L) values for each mechanism. These values are then put in the numerator of the Equation 5.1. Table 5.6 then shows security index value of each security mechanism. Higher the value of security index, higher is the efficiency of the algorithm, and the selection of that algorithm is more probable. Algorithms having value of security index less than '0.5' should not be selected as security algorithms for any system. Therefore, the choice should be done among the algorithms having security index greater than or equal to '0.5'.

$$\text{Security Index} = \frac{(3*H) + (2*M) + (1*L)}{(3*19) + (2*13) + (1*32)} \quad (5.1)$$

Each security algorithm is indexed on the basis of the security requirements fulfilled by it. Security requirements are marked with number of high, medium and low risk threats. The prioritized threats are considered so that the algorithms which mitigate the higher number of high priority threats are indexed higher than the algorithms which mitigate the number of low priority threats.

For example, Algorithm A mitigates 3 threats (3,0,0) and Algorithm B also mitigates 3 threats (0,0,3). Now, if we simply consider the number of threats mitigated divided by the total number of threats, then Algorithm A and B have same security index as 0.026 (3/115). But as per the proposed method, Algorithm A has security index value 0.079 and B has 0.026.

Table 4.4: Security Requirements and Algorithms mapping

Mechanism Requirements	AES	DES	ECC	HECC	ECDSA	HECDSA	SHA1	ECC + DUAL RSA + MD5	ECIES
Identification	N	N	Y (2,0,1)	Y (2,0,1)	Y (2,0,1)	Y (2,0,1)	N	Y (2,0,1)	Y (2,0,1)
Authentication	N	N	Y (4,1,3)	N	N	N	N	Y (4,1,3)	Y (4,1,3)
Authorization	N	N	N	N	Y (6,1,3)	Y (6,1,3)	N	Y (6,1,3)	Y (6,1,3)
Intrusion Detection	N	N	Y (1,1,2)	Y (1,1,2)	Y (1,1,2)	Y (1,1,2)	Y (1,1,2)	Y (1,1,2)	Y (1,1,2)
Immunity	Y (0,0,4)	Y (0,0,4)	Y (0,0,4)	Y (0,0,4)	N	N	N	Y (0,0,4)	Y (0,0,4)
Integrity	N	N	N	N	N	N	Y (0,0,2)	Y (0,0,2)	Y (0,0,2)
Privacy	Y (0,3,0)	Y (0,3,0)	Y (0,3,0)	Y (0,3,0)	N	N	N	Y (0,3,0)	Y (0,3,0)
Physical Protection	Y (0,0,2)	N	N	N	N	N	N	Y (0,0,2)	Y (0,0,2)
Trust	N	N	N	N	N	N	N	N	Y (7,5,10)
Survivability	N	N	N	N	N	N	N	N	N
System Maintenance	N	N	N	N	N	N	N	N	N
Total (19,13,32)	(0,3,6)	(0,3,4)	(7,5,10)	(3,4,7)	(9,2,6)	(9,2,6)	(1,1,4)	(13,6,17)	(18,11,27)

Chapter 5

Case study- A Smart Home

Till now the chapters have discussed the security engineering framework for generic home automation systems. In this chapter, we define a basic smart home architecture. This smart home is considered as a case study and results of our structured approach are shown.

5.1 Smart Home model

Every IoT application aims at providing ubiquitous and remote connectivity. Smart Homes open up the gates for remote controlling and supervision of homes. It also provides comfort by automatically switching on the lights when required or by adjusting the temperature of the room.

The proposed model for smart home consists mainly of smart devices, various sensors, and RFID tags. Each of these devices is controlled by their specific controllers, and these distributed controllers are controlled and managed by the central home gateway system. Components communicate with the central gateway via their respective controllers only, i.e. they follow a hierarchy. Users can control the home via web/mobile interfaces.

5.1.1 Important devices and communication flow

This section discusses the different devices used in the smart home systems. Also, it describes how the devices communicate with one another. The system is divided into three layers on the basis of communication flow. Figure 5.1 shows how the communication takes place between the different layers.

Smart Devices: Smart devices are the digitally advanced devices which train and adapt themselves according to the behavior of the owner. This training and adaptation are possible because of the underlying machine learning techniques. For example, a smart AC turns on automatically whenever it senses someone in the room. It also senses where the person is and then targets wind in that direction. One can turn on the AC, thermostat, heater, etc. on his way home. Smart AC, smart TV, refrigerator, washing machine and security locks are some of the most popularly used smart devices in a home automation system.

Sensors: Sensors are designed to sense a specific parameter from the environment or surroundings. Different sensors being used in the system are: Light sensors, temperature sensors, motion sensors and smoke sensors. Motion sensors keep track of every activity happening in the house. Door sensors will sense the closing and opening of the door. Light sensors detect the intensity of the light in the room and brighten accordingly.

RFID tags: RFID tags are small chips which are attached to the items for tracking and identification purposes. In homes, sometimes we keep something at a place and then forget about where we have kept it. If the item has RFID tag attached to it, then he/she can locate the item with the help of RFID controller.

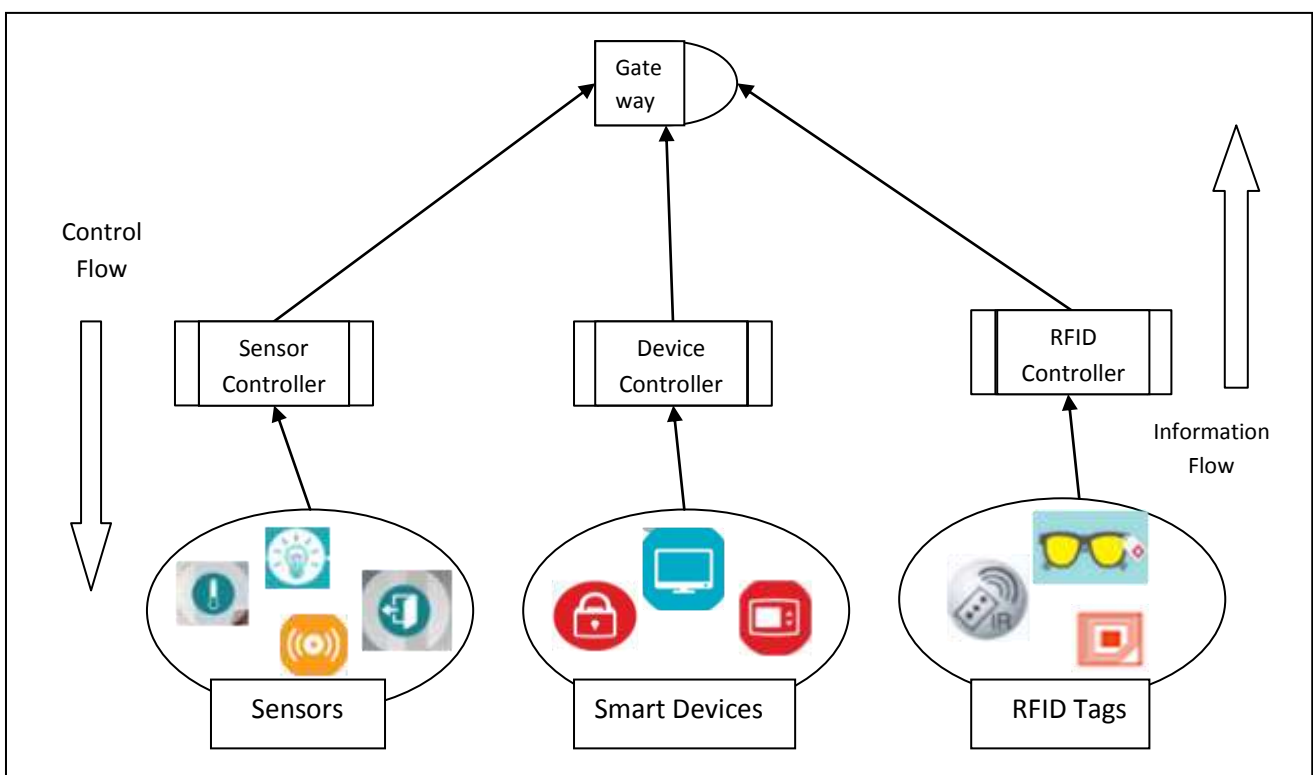


Figure 5.1: Communication and information flow in smart home network

Device controller: Device controller maintains all the smart devices within the system. They maintain a list of all the smart devices that comes under their supervision, checks the working status (faulty or not, working or not, on or off) of these devices. These devices are connected to the gateway via the controller.

Sensor controller: Sensor controller monitors the different sensors existing in the system. It is responsible for providing coordination among the different sensors for a smooth and efficient

functioning. Sensors send data to the controller which then decides whether to act or to send data to the cloud. Sensors can sense, but they actuate according to the controller.

RFID controller: RFID tags are small chips which can be identified on the basis of their unique ID. These tags are sensed and located by a RFID reader. RFID controller is the RFID reader, and it also maintains a database for the home user. This database maps unique RFID number to a custom name given by the user. This mapping helps the user to find the item he is looking for and saves him from the effort of remembering a lot of numbers.

Central Home Gateway: Central home gateway connects the home to the internet. It acts like a bridge taking care of all of the communication going on in the network. It makes the remote access of the home possible.

Table 5.1 briefly describes the different hierarchical layers, their components and working of the components.

Table 5.1: Layer wise components of smart home

Lower Layer	Devices	Sensors	RFID tags
	These devices can sense and actuate. These have digital displays and learn the behavior of the user based on machine learning techniques and are called smart devices. Smart TV, Smart Refrigerator, Smart AC, Smart washing machine, security locks	Sensors sense the environment based on their specified properties. They can send the data. Light sensor, temperature sensor, motion sensor, smoke sensor, door sensor	RFID tags are important to know where the thing is. Important files and most commonly misplaced things like scissors etc. can be tagged, for tracking purpose.
These are the basic devices and form the lower layer of the model.			
Middle layer	Device controller	Sensor controller	RFID controller
	It controls the working and functioning of the smart devices. The central home gateway communicates with the devices via this controller. Controllers are responsible for detecting and reporting of failure of any device.	Sensors send their data to the sensor controller. The controller also keeps a check on the status of the sensors, whether working or not.	This controller maintains a record of every available tag in the system. It keeps an index of Unique Id and custom names of the tag.
These controllers work as a medium between the actual components and the central gateway. Controllers are responsible for reporting of failure or misbehave of any device under their domain.			
Upper Layer	Central home gateway		
	This gateway acts as an interface between the home network and the outside world. This gateway allows remote access. On one end it is connecting to the internet, and on other, it is connected to the home controllers.		

5.1.3 Case Study

In a smart home environment, one can access the automated devices at home via smart phone application or web interface. Users can control the devices even if they are away from home. The smart home aims at making life easy and more comfortable. Figure 5.2 shows the positioning of different devices and sensors within the home. The central home gateway controls the communication to and from the home system. The Internet connects the remote user with the home.

The previous section discusses the devices and sensors present in the home system. This section shows the positioning of the different devices in a home environment.

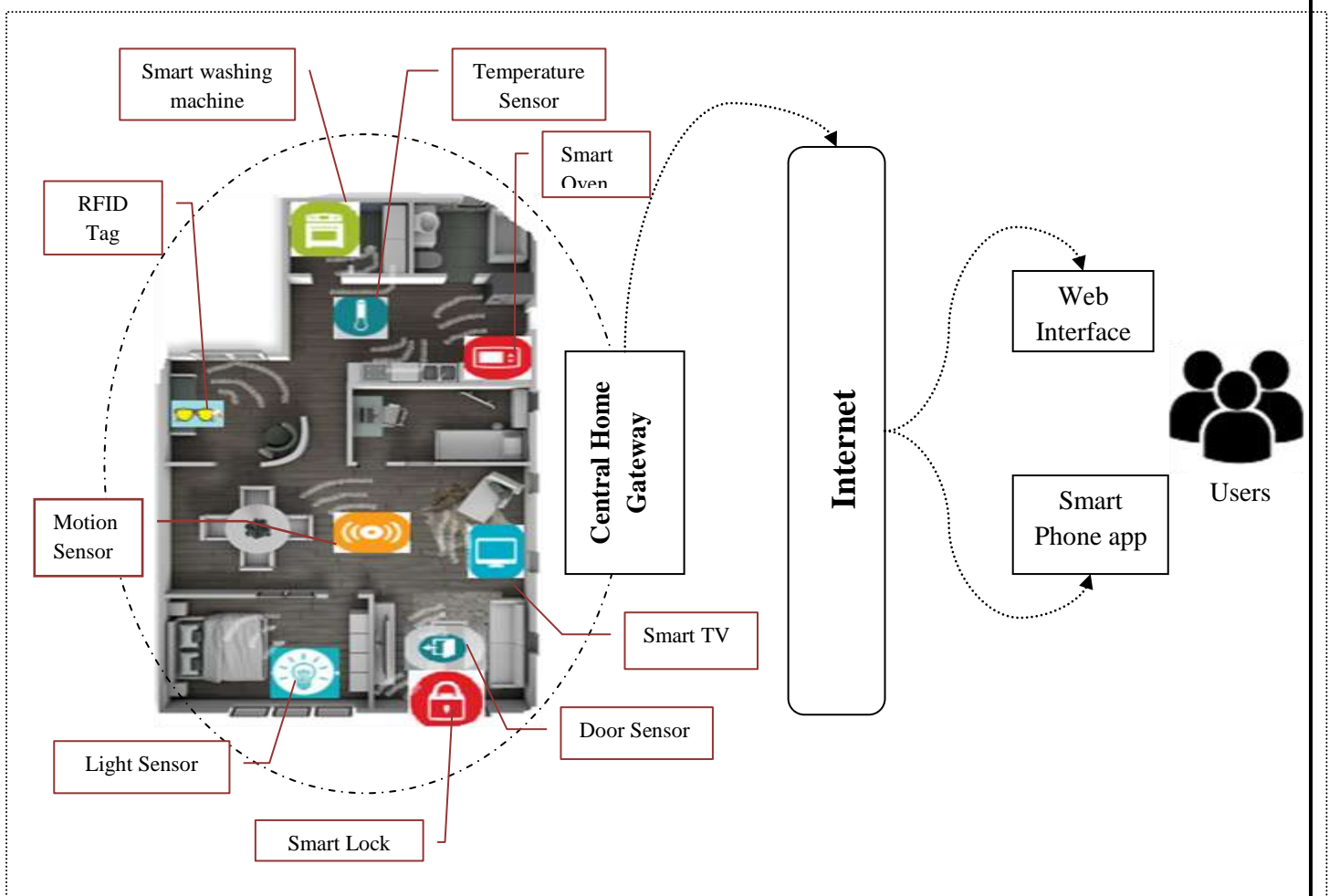


Figure 5.2 Overall architecture of Smart Home automation

5.2 Security Requirements Engineering

As in section 4.2, Requirements are elicited for generic home automation system. In this section, the requirements are elicited for the specific smart home mentioned in section 5.1.

5.2.1 Requirements Elicitation

Actors, assets, vulnerabilities, threats and security requirements are identified.

- **Identification of actors:** Actors for this case study are identified on the basis of the actors identified in section 3.2.1. Table 5.2 depicts the different actors of the system.

Table 5.2: Actors for Smart Home

Users	Owner, guests, inhabitants, visitor
Communication Channel	Internet
Interfaces	Laptop, computer, mobile phone
End Point Applications	Website, phone application
Gateway	Central Home Gateway
Devices	Light sensors, temperature sensors, motion sensors,

- **Identification of assets:** The different assets of the case study are depicted in Table 5.3. These assets are important for the system. Assets should be protected against unauthorized access or physical theft as they are linked to the actors.

Table 5.3: Assets for the case study

System Data	User preferences, users' habits, media (photos, recordings)
Privacy	Privacy of users, communication privacy
Logs	Network logs, access logs
Network	Internet, WSN, RSN, Bluetooth
Trust	Untrustworthy guest, faulty component
Devices	Sensors, home appliances, home utilities
Data storage units	Cloud, Data servers

- **Identification of vulnerability:** Following vulnerabilities are identified for the system based on Table 3.2. Naive users, Weak Access Control, Unencrypted data, Insecure networks, Obsolete systems, Interoperability Issues, Physical security,

Remote access, Malware attack, Information leakage, Intrusion detection, Obsolete data, Inaccurate data, Device/system failure, Inefficient security configurations, Eavesdropping/ resource isolation, Insecure interfaces, Insecure network services, Improper/inefficient logging, Misconfiguration, Lack of standards, Firewall inefficiency, Monitoring absence, System's resources misuse, Weak cryptographic techniques, and Unauthorized access.

- **Identification of threats:** Following threats are identified for the system based on Table 3.3. Fraud, Data theft, Identity theft, Credential theft, Spoofing, Phishing, DDoS, Malware attack, Security breach, Unavailability, System failure, Hardware failure, Software crash, Eavesdropping, Privacy violation, System misuse, Technical failure, Power failure, Flooding attack, Node attack, Node capture, Change data, Man in the middle attack, Communication change, Data leakage, and Routing attack.
- **Identification of security requirements:** The security requirements identified for the system are: Identification, Authentication, Authorization, Privacy, Immunity, Intrusion detection, Physical Security, Security Maintenance, Integrity, and Trust.

5.2.2 Requirement Analysis

The requirements identified for the case study are similar to the requirements as identified in section 3.2.1. Therefore, the same mappings of section 3.2.2 and tables 3.5, 3.6, 3.7 and 3.8 can be used here as well.

5.2.3 Risk Calculation

On the basis of threat rating and impact rating, risk value is calculated for each threat. Table 3.13 depicts the risk values for the threats.

5.2.4 Threat Prioritization

Based on the Risk values obtained in Table 3.13, threats are prioritized in three categories: Low risk (Risk value < 35), medium risk ($35 \leq$ Risk value < 60), and high risk ($60 \leq$ Risk Value < 100). Table 3.14 depicts this prioritization.

5.3 Security Design Engineering

This section discusses the steps followed in the design phase and shows the result.

5.3.1 Security Requirements Risk Evaluation

Based on the method discussed in section 4.1.1, the triplet for security requirements is showed in Table 5.4.

Table 5.4: Risk values for Security Requirements

Security Requirements	Risk Values
Identification	(2,0,1)
Authentication	(4,1,3)
Authorization	(6,1,3)
Immunity	(0,0,4)
Integrity	(0,0,2)
Intrusion detection	(1,1,2)
Privacy	(0,3,0)
System Maintenance	(1,1,3)
Physical protection	(0,0,2)
Trust	(7,5,10)

5.3.2 Security Requirements and Security Services

In this section, we consider only confidentiality and trust security services. These security services are provided by cryptographic algorithms.

5.3.3 Creation of Security Algorithms Repository

Out of the available cryptographic algorithms, following algorithms are chosen for this system: AES, DES, ECC, HECC, SHA1, ECDSA, HECDISA, ECC+DUAL RSA+MD5, ECIES.

Design Constraints for Smart Home

Design constraints for smart home devices are discussed in Table 5.5. The values for these constraints are set low, medium and high. These values are assigned to the devices on the basis of IoT architecture mentioned in section 2.1.

Table 5.5: IoT constraints and their specifications for security algorithms

Constraints	Perception Layer	Transportation Layer	Application Layer
Power	Low	Medium	High
Memory	Low	Low-Medium	High
Embedded	Yes	Maybe	Maybe
Processing	Low	Medium	High
Mobility	Low	Medium	Medium-High

5.3.4 Calculation of Security Index

For the algorithms chosen in section 5.3.3, Security Index is calculated for each of these algorithms. Table 5.6 depicts the security index for the security algorithms. This security index is calculated on the basis of Table 4.5.

Table 5.6: Security Index of algorithms

Security Mechanism	Security Index
AES	0.104
DES	0.086
ECC	0.3565
HECC	0.20869
ECDSA	0.3217
HECDSA	0.3217
SHA1	0.0782
ECC + DUAL RSA + MD5	0.59130
ECIES	0.8956

Based on this calculation, we can say that ECIES is the best available security algorithm which if implemented on this system will solve 89.56% of the security issues. Depending on the security requirements as demanded by various users, we can suggest different algorithms based on this mapping and the security index value.

This chapter discusses the implementation of the security engineering framework tool. The tool is designed on the basis of the methodology applied in this thesis. This chapter shows the use of the tool with respect to the IoT system of home automation described in chapter 3. The use of the tool is not limited to this application only; it can work well for other IoT-based systems as well with a little or no modification.

6.1 Introduction

An IoT-based smart home automation system is taken as a case study for giving a clear understanding of the applied framework. The tool for applying this security engineering framework is made using MATLAB 2014a on a Windows PC with 8 GB RAM. The interface consists of six tabs which cover all phases of the framework. The last tab labeled as "6-Security Index" shows the security index for each algorithm. This result will help in choosing the optimal algorithm for implementation so that maximum security is assured. Figure 6.1 shows the home/starting screen of the tool.



Figure 6.1: Home screen of the tool

6.2 Working of the Security Engineering Framework for IoT

This subtopic discusses the working of the developed tool. The tool has six different tabs. First four tabs take care of the steps involved in Security Requirements Elicitation and Analysis phase. The last two tabs take care of the Security Design Engineering phase. Figure 6.2 shows the second screen of the tool which consists of all the tabs and the layout of the home model.



Figure 6.2: Second screen with tabs for each phase

6.2.1 Phase I: Security Requirements Engineering

The first tab takes care of requirements elicitation. Figure 6.3 shows how the actors, assets, vulnerabilities, and threats are added for a system.

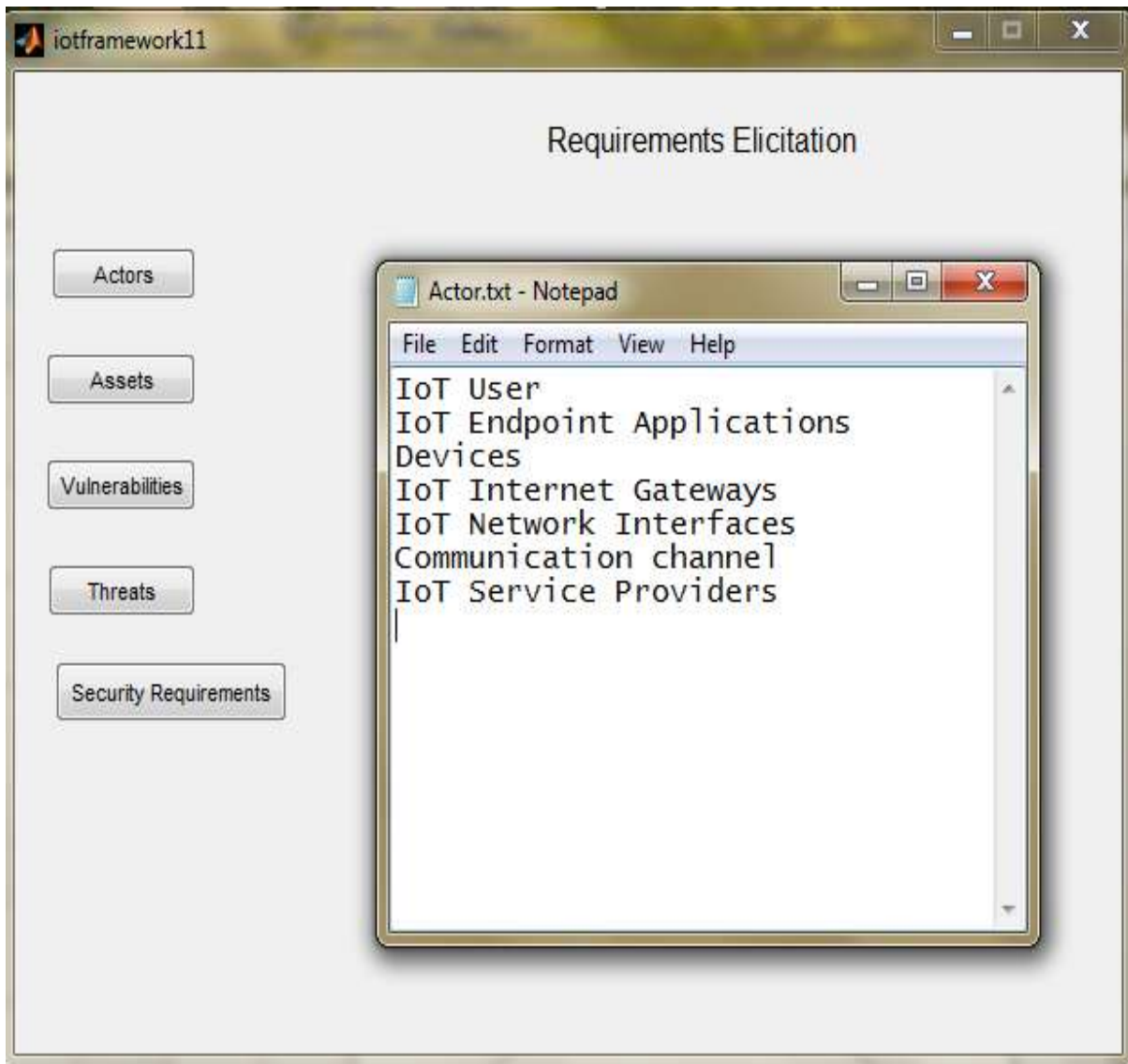
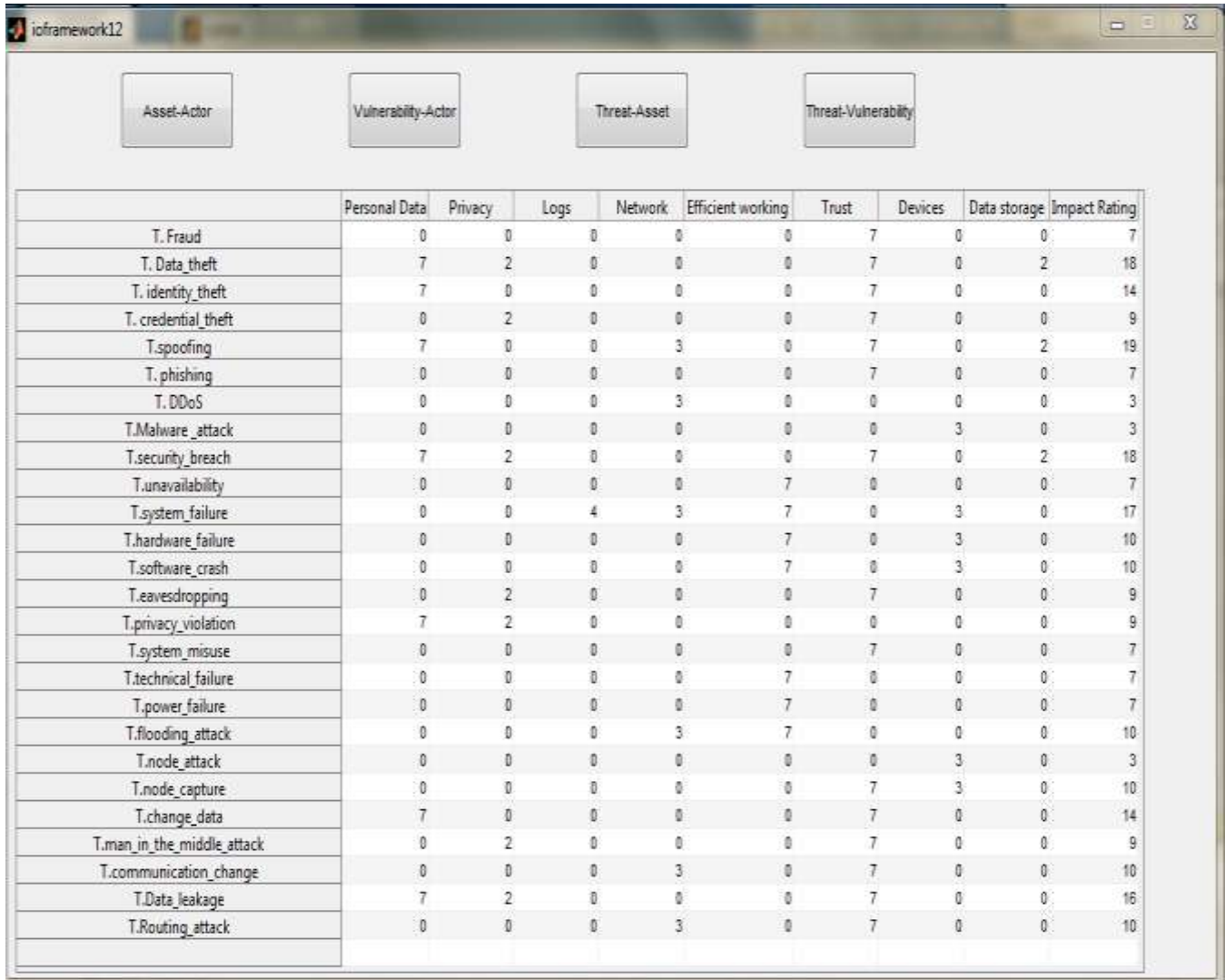


Figure 6.3: Tab 1- Requirements Elicitation

Once elicitation is done, the mapping between the different factors is done. Figure 6.4 shows the mapping between assets and threats. This mapping gives Impact Rating, which is then used to calculate risk. Tab 2 has four different options, one for each mapping.

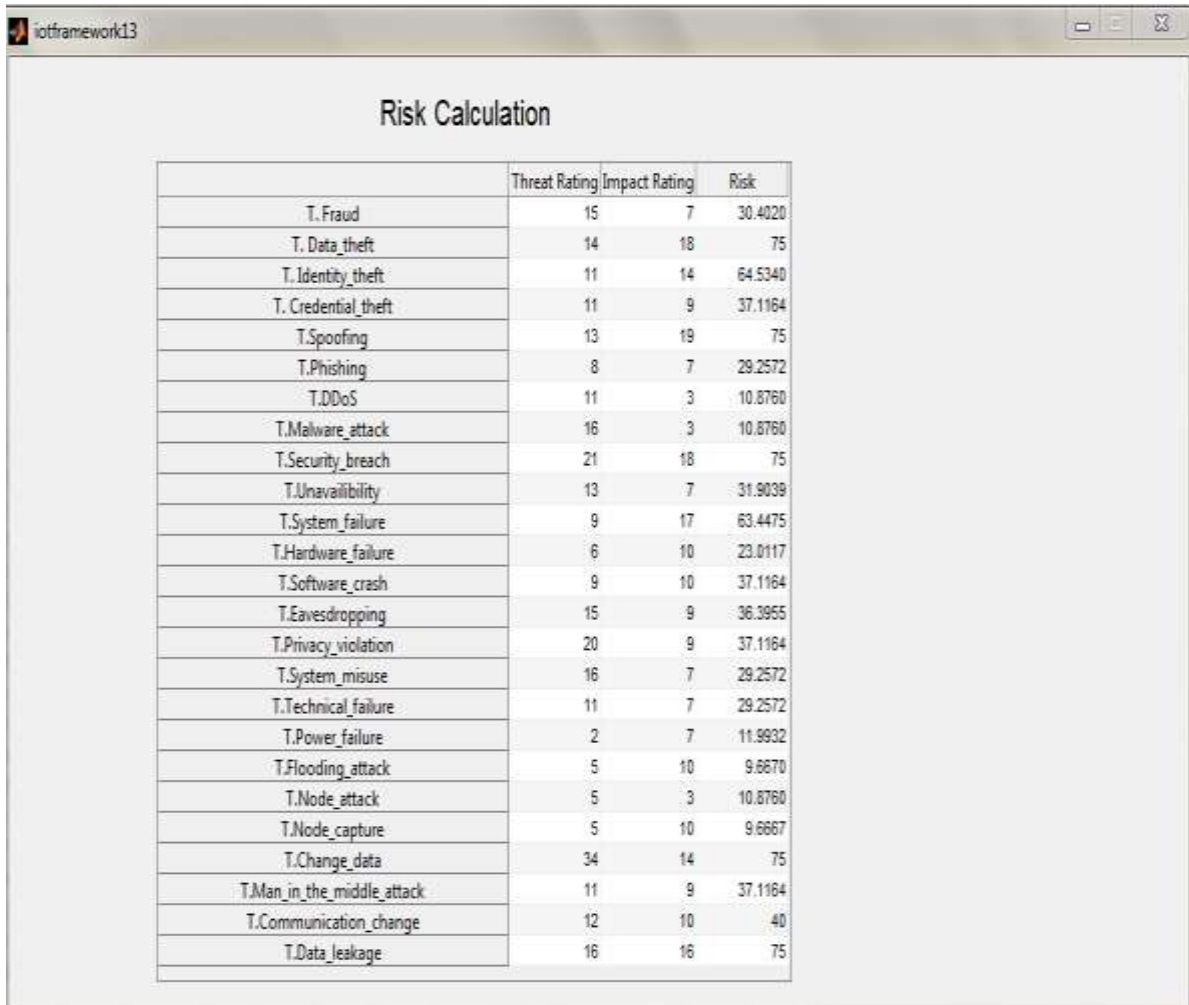


The screenshot shows a software interface titled 'ioframework12'. At the top, there are four buttons: 'Asset-Actor', 'Vulnerability-Actor', 'Threat-Asset', and 'Threat-Vulnerability'. Below these buttons is a table with 11 columns: 'Personal Data', 'Privacy', 'Logs', 'Network', 'Efficient working', 'Trust', 'Devices', 'Data storage', and 'Impact Rating'. The table contains 25 rows of threat data, including 'T. Fraud', 'T. Data_theft', 'T. identity_theft', 'T. credential_theft', 'T. spoofing', 'T. phishing', 'T. DDoS', 'T. Malware_attack', 'T. security_breach', 'T. unavailability', 'T. system_failure', 'T. hardware_failure', 'T. software_crash', 'T. eavesdropping', 'T. privacy_violation', 'T. system_misuse', 'T. technical_failure', 'T. power_failure', 'T. flooding_attack', 'T. node_attack', 'T. node_capture', 'T. change_data', 'T. man_in_the_middle_attack', 'T. communication_change', 'T. Data_leakage', and 'T. Routing_attack'.

	Personal Data	Privacy	Logs	Network	Efficient working	Trust	Devices	Data storage	Impact Rating
T. Fraud	0	0	0	0	0	7	0	0	7
T. Data_theft	7	2	0	0	0	7	0	2	18
T. identity_theft	7	0	0	0	0	7	0	0	14
T. credential_theft	0	2	0	0	0	7	0	0	9
T. spoofing	7	0	0	3	0	7	0	2	19
T. phishing	0	0	0	0	0	7	0	0	7
T. DDoS	0	0	0	3	0	0	0	0	3
T. Malware_attack	0	0	0	0	0	0	3	0	3
T. security_breach	7	2	0	0	0	7	0	2	18
T. unavailability	0	0	0	0	7	0	0	0	7
T. system_failure	0	0	4	3	7	0	3	0	17
T. hardware_failure	0	0	0	0	7	0	3	0	10
T. software_crash	0	0	0	0	7	0	3	0	10
T. eavesdropping	0	2	0	0	0	7	0	0	9
T. privacy_violation	7	2	0	0	0	0	0	0	9
T. system_misuse	0	0	0	0	0	7	0	0	7
T. technical_failure	0	0	0	0	7	0	0	0	7
T. power_failure	0	0	0	0	7	0	0	0	7
T. flooding_attack	0	0	0	3	7	0	0	0	10
T. node_attack	0	0	0	0	0	0	3	0	3
T. node_capture	0	0	0	0	0	7	3	0	10
T. change_data	7	0	0	0	0	7	0	0	14
T. man_in_the_middle_attack	0	2	0	0	0	7	0	0	9
T. communication_change	0	0	0	3	0	7	0	0	10
T. Data_leakage	7	2	0	0	0	7	0	0	16
T. Routing_attack	0	0	0	3	0	7	0	0	10

Figure 6.4: Tab-2 Requirement Analysis

Figure 6.5 shows the values of risk calculated for each threat based on the Threat rating and Impact rating. After step 2, i.e. analysis, the tool in the backend runs fuzzy inference system CalculateRisk.fis and generated the table as shown in Figure 6.5 after this calculation.



The screenshot shows a window titled 'Risk Calculation' with a table containing 20 rows of threat data. Each row lists a threat type, its Threat Rating, Impact Rating, and a calculated Risk value. The Risk values are numerical, some with decimal parts, and are sorted in descending order from top to bottom.

	Threat Rating	Impact Rating	Risk
T. Fraud	15	7	30.4020
T. Data_theft	14	18	75
T. Identity_theft	11	14	64.5340
T. Credential_theft	11	9	37.1164
T.Spoofing	13	19	75
T.Phishing	8	7	29.2572
T.DDoS	11	3	10.8760
T.Malware_attack	16	3	10.8760
T.Security_breach	21	18	75
T.Unavailability	13	7	31.9039
T.System_failure	9	17	63.4475
T.Hardware_failure	6	10	23.0117
T.Software_crash	9	10	37.1164
T.Eavesdropping	15	9	36.3955
T.Privacy_violation	20	9	37.1164
T.System_misuse	16	7	29.2572
T.Technical_failure	11	7	29.2572
T.Power_failure	2	7	11.9932
T.Flooding_attack	5	10	9.6670
T.Node_attack	5	3	10.8760
T.Node_capture	5	10	9.6667
T.Change_data	34	14	75
T.Man_in_the_middle_attack	11	9	37.1164
T.Communication_change	12	10	40
T.Data_leakage	16	16	75

Figure 6.5: Tab-3 Risk Calculation

After risk calculation, threats are prioritized as Low, Medium, High. This prioritization is shown in Figure 6.6

Threat Prioritization

	High	Medium	Low
1	T.Data_leakage	T.Communication_change	T.Fraud
2	T.Change_data	T.Credential_theft	T.Phishing
3	T.System_failure	T.Software_crash	T.DDoS
4	T.Spoofing	T.Eavesdropping	T.Malware_attack
5	T.Identity_theft	T.Privacy_violation	T.Unavailability
6	T.Data_theft	T.Man_in_the_middle_attack	T.Hardware_failure
7	T.Security_breach		T.System_misuse
8			T.Technical_failure
9			T.Power_failure
10			T.Flooding_attack
11			T.Node_attack
12			T.Node_capture

Figure 6.6: Tab 4-Threat Prioritization

6.2.2 Phase II: Security Design Engineering

Security requirements and threats mapping is done in Phase I. Based on this mapping, risk value of each security requirement is calculated as a triplet of $\langle H, M, L \rangle$, where H, M, L is the number of high, medium and low priority threats which are mitigated, if that security requirement is fulfilled. The last two tabs does the algorithms mapping with security requirements and calculates security index for every selected algorithm.

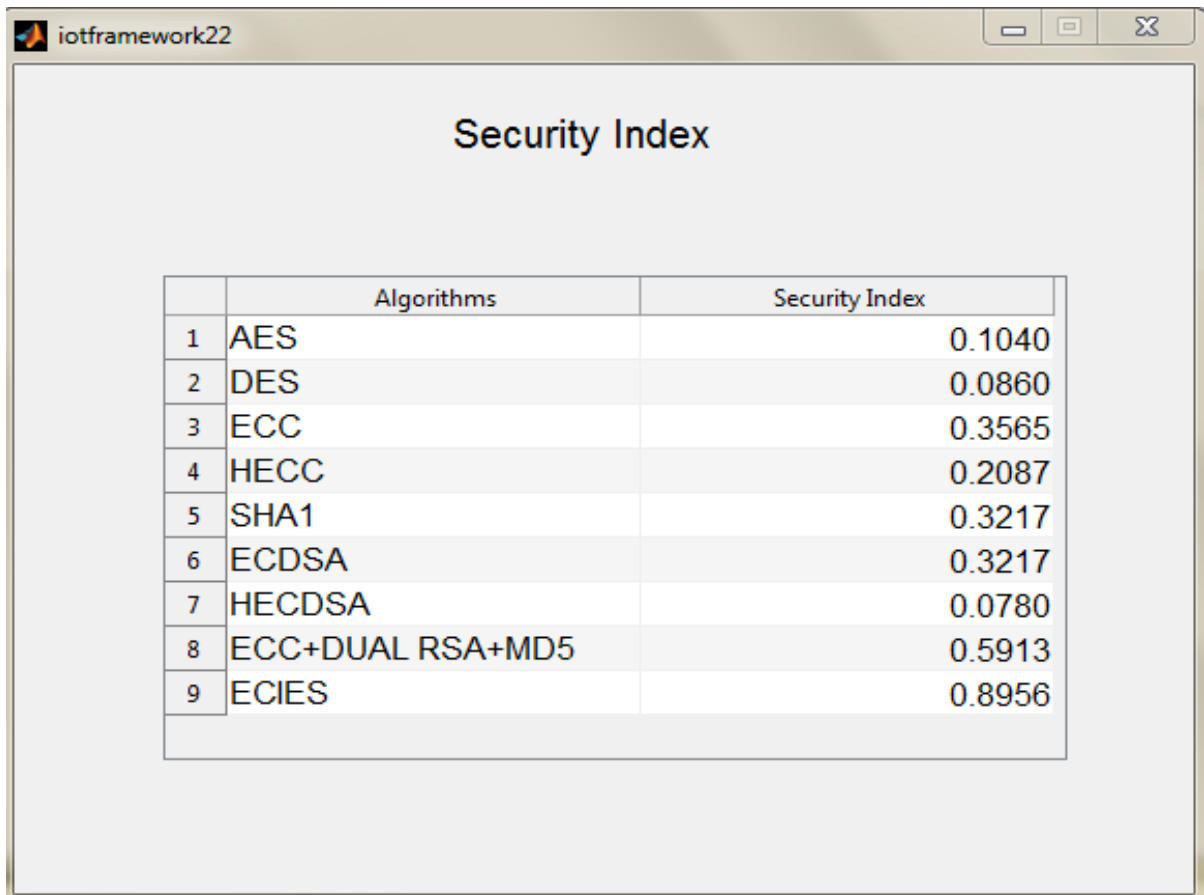
Figure 6.7 shows the mapping of security algorithms and the security requirements they fulfill. Based on this step, security index is calculated.

The screenshot shows a window titled "Security requirements and Security Algorithms mapping" with a table containing the following data:

	AES	DES	ECC	HECC	ECDSA	HECDSA	SHA1	ECC+DUAL RSA+MD5	ECIES
IDENTIFICATION	N	N	Y	Y	Y	Y	N	Y	Y
AUTHENTICATION	N	N	Y	N	N	N	N	Y	Y
AUTHORIZATION	N	N	N	N	Y	Y	N	Y	Y
INTRUSION DETECTION	N	N	Y	Y	Y	Y	Y	Y	Y
IMMUNITY	Y	Y	Y	Y	N	N	N	Y	Y
INTEGRITY	N	N	N	N	N	N	Y	Y	Y
PRIVACY	Y	Y	Y	Y	N	N	N	Y	Y
PHYSICAL PROTECTION	Y	N	N	N	N	N	N	Y	Y
TRUST	N	N	N	N	N	N	N	N	Y
SURVIVABILITY	N	N	N	N	N	N	N	N	N
SYSTEM MAINTENANCE	N	N	N	N	N	N	N	N	N

Figure 6.7: Tab 5-Security Requirement mapping

Security index is calculated for each algorithm based on the mapping obtained in Tab 5-Security Req. Elicitation and using the formula for security index as mentioned in Equation 5.1 in Chapter 5. Figure 6.8 shows the calculation. The value of security index lies [0,1]. Higher the value, higher the efficiency of that algorithm and vice-versa.



The screenshot shows a window titled 'iotframework22' with a 'Security Index' tab. The window contains a table with the following data:

	Algorithms	Security Index
1	AES	0.1040
2	DES	0.0860
3	ECC	0.3565
4	HECC	0.2087
5	SHA1	0.3217
6	ECDSA	0.3217
7	HECDSA	0.0780
8	ECC+DUAL RSA+MD5	0.5913
9	ECIES	0.8956

Figure 6.8: Tab 6-Security Index

This chapter concludes the work and gives insight into the future work.

7.1 Conclusions

IoT is seamlessly integrating the physical and virtual things into information network. IoT's realization depends on continues technical innovation in a number of fields. The security of IoT is very important. Since, IoT is in its growing phase. For its positive growth, the shortcomings have to be removed. Security, privacy and trust has to be provided to gain the confidence of the users. Every solution has its strengths and limitations. New techniques should be discovered which can give maximum throughput and have minimum limitations. The security solutions should be dynamic and adaptive.

Based on the well-defined steps present in the proposed Security Engineering Framework, security requirements elicitation is done by identifying actors, their functional and non-functional requirements, identifying vulnerabilities, assets and all possible attacks on them and then they were mapped with security requirements. Then using the fuzzy logic, risk was calculated for each threat. Based on the risk assessment values, threats are prioritized. This requirements analysis is done for home automation systems.

In security design engineering, some security objectives were first identified namely Confidentiality, Integrity, Authentication, Authorization and Trust to map security requirements. Security requirements are then mapped with the security services. For IoT system this work emphasized that not only cryptographic techniques but additional techniques using same needs to be in place. Algorithms meeting all above mentioned security requirements were listed. Security index is calculated for each algorithm, so that, for implementing security, the algorithms are not selected on ad-hoc basis.

There is no accepted definition of security requirements, however all the work in this thesis shows that Security Requirements Engineering is really important as part of software engineering development cycle for making IoT application robust, more secure and reliable. If the phases described in this methodology are properly implemented it will assist the software engineering team to make correct decisions to overcome most of the known security threats to the system. It is also clear that there is inadequacy of security in IoT systems. It

may seem good enough but it will cost more if these are not dealt in early stages of the development cycle.

The framework and the new methodology proposed in this thesis can be adopted as a generic model for enhancing security in many IoT Applications, as in case study it was shown how we can achieve it on " a smart home model". This thesis work also provided an insight about the challenges and areas of the Internet of Things technology which can help in its further improvement and hence contribute to the betterment of our society.

So, it is concluded that the work a fuzzy based methodology to implement security in IoT provided certain ground breaking improvements and suggestions in the field of IoT security. This thesis also indicates that IoT lacks security and a lot can be done in the field of security for IoT-based systems.

7.2 Future Work

This thesis applied a security engineering framework to the smart home. The future directions of this work are as followed:

- This model can be applied to other applications such as smart grid, smart city, smart health, smart energy, smart transport and others.
- This framework can further be expanded by implementing the algorithms and testing the system security level. If any security requirement is not fulfilled, then one can try to find out a way to ensure complete security, by developing a hybrid algorithm.
- Security of IoT can be ensured by trusted real-time models. This field can also be explored for coming up with better security solutions. New encryption and key management techniques can be discovered.
- Energy efficient hardware and software security solutions can be proposed.
- Machine learning techniques can be used to automate the mapping process.

References

- [1] Internet of Things (IoT) History (Feb 1,2016). Retrieved from <https://www.postscapes.com/internet-of-things-history/>
- [2] Gartner Says 6.4 billion Connected "Things" Will Be in Use in 2016, Up 30 percent From 2015. Retrieved from www.gartner.com/newsroom/id/3165317
- [3] Rosslin John Robles, Tai-hoon Kim, "A Review on Security in Smart Home Development", International Journal of Advanced Science and Technology Vol. 15, February, 2010
- [4] Nikos Komninos, Andreas Pitsillides, " Survey in Smart Grid and Smart Home Security: Issues, Challenges and Countermeasures", IEEE COMMUNICATION SURVEYS & TUTORIALS, VOL. 16, NO. 4, FOURTH QUARTER 2014
- [5] Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, Imrich Chlamtac, "Internet of things: Vision, applications and research challenges",D. Miorandi et al. / Ad Hoc Networks 10 (2012) 1497–1516
- [6] Kai Zhao, LinaGe, "A Survey on the Internet of Things Security". IEEE 2013 Ninth International Conference on Computational Intelligence and Security.
- [7] Qi Jing, Athanasios V, Vasilako, Jiafu Wan, Jingwei Lu, Dechao Qiu, "Security of the Internet of Things: perspectives and challenges", Wireless Netw (2014) 20:2481–2501
- [8] Arbia Riahi, Yacine Challal, Enrico Natalizio, Zied Chtourou, Abdelmadjid Bouabdallah," A systemic approach for IoT security". 2013 IEEE International Conference on Distributed Computing in Sensor Systems
- [9] Antonio J. Jara, Miguel A. Zamora and Antonio F. G. Skarmeta, "An architecture based on Internet of Things to support mobility and security in medical environments" , IEEE CCNC 2010 proceedings.
- [10] Liang Zhou, Han-Chieh Chao," Multimedia Traffic Security Architecture for the Internet of Things"IEEE Network ,May/June 2011 0890-8044/11. 35-40pp
- [11] Thomas Kothmayr, Corinna Schmitt, Wen Hu, Michael Brünig and Georg Carle, "A DTLS Based End-To-End Security Architecture for the Internet of Things with Two-Way Authentication", 2012 IEEE.
- [12] Quangang Wen, Xinzheng Dong, Ronggao Zhang, "Application Of Dynamic Variable Cipher Security Certificate In Internet Of Things" Proceedings of IEEE CCIS2012
- [13] Wang Chen, "An Ibe-Based Security Scheme On Internet Of Things". 2012 IEEE

- [14] Siwei Peng, " An Id-Based Multiple Authentication Scheme Against Attacks In Wireless Sensor Networks". Proceedings of IEEE CCIS2012. pp 1042-1045
- [15] Zhihua Li, Xi Yin, Zhenmin Geng, Haitao Zhang, Pengfei Li, Ya Sun, Huawei Zhang, Lin Li, "Research on PKI-like Protocol for the Internet of Things",2012 IEEE DOI 10.1109/ICMTMA.2013.227
- [16] Shahid Raza, Linus Wallgren, Thiemo Voigt, " SVELTE: Real-time intrusion detection in the Internet of Things". Ad Hoc Networks 11 (2013) 2661–2674
- [17] Kakali Chatterjee, Asok De, Daya Gupta; " Mutual Authentication Protocol Using Hyperelliptic Curve Cryptosystem in Constrained Devices ". International Journal of Network Security, Vol.15, No.1, PP.9-15, Jan. 2013
- [18] Teng Xu, James B. Wendt, and Miodrag Potkonjak " Security of IoT Systems: Design Challenges and Opportunities ".2014 IEEE
- [19] Savio Sciancalepore, Giuseppe Piro, Elvis Vogli, Gennaro Boggia, and Luigi Alfredo Grieco, "On securing IEEE 802.15.4 networks through a standard compliant framework"
- [20] Somia Sahraoui, Azeddine Bilami, " Compressed and Distributed Host Identity Protocol for End-to-End Security in the IoT " 2014 Fifth International Conference on Next Generation Networks and Services (NGNS) May 28-30, 2014, Casablanca, Morocco
- [21] Kai Fan, Chen Liang, Hui Li, Yintang Yang, " LRMAPC: a lightweight RFID mutual authentication protocol with cache in the reader for IoT ". 2014 IEEE
- [22] Jun-Ya Lee, Wei-Cheng Lin, Yu-Hung Huang; " A Lightweight Authentication Protocol for Internet of Things " 2014, IEEE
- [23] Shohreh Hosseinzadeh, Sampsa Rauti, Sami Hyrynsalmi, Ville Leppänen; " Security in the Internet of Things through Obfuscation and Diversification " 2015 IEEE
- [24] Kai Fan, Yuanyuan Gong, Zhao Du, Hui Li, Yintang Yang; "RFID secure application revocation for IoT in 5G", Published in 2015 IEEE Trustcom/BigDataSE/ISPA. pp 175-181
- [25] Ismail Butun, Burak Kantarci, Melike Erol-Kantarci, "Anomaly detection and privacy preservation in Cloud-Centric Internet of Things" IEEE ICC 2015 - Workshop on Security and Privacy for Internet of Things and Cyber-Physical Systems. pp 2610-2615

- [26] Mary R. Schurgot, David A. Shinberg, Lloyd G. Greenwald. "Experiments with Security and Privacy in IoT Networks" Published in 2015 IEEE. pp 1-6
- [27] Ricardo Neisse, Gary Steri, Igor Nai Fovino, Gianmarco Baldini. "SecKit: A Model-based Security Toolkit for the Internet of Things". *computers & security* 54(2015)60-76
- [28] Ikram Ullah, Munam Ali Shah. " A Novel Model for Preserving Location Privacy in Internet of Things". *IEEE Conference Publications 2016*, pp 542 - 547
- [29] M. Surendar, A. Umamakeswari. " InDReS: An Intrusion Detection and Response System for Internet of Things with 6Lo WP AN" Published in IEEE WiSPNET 2016 conference. pp 1903-1908
- [30] Nouha Oualha, Kim Thuat Nguyen. " Lightweight Attribute-based Encryption for the Internet of Things". 2016 IEEE. pp 1-6
- [31] Manju Suresh, Neema M. " Hardware implementation of blowfish algorithm for the secure data transmission in Internet of Things". *Global Colloquium in Recent Advancement and Effectual Researches in Engineering, Science and Technology (RAEREST 2016)*. *Procedia Technology* 25 (2016) 248 – 255
- [32] Sanaz Rahimi Moosavia, Tuan Nguyen Gia, Ethiopia Nigussie, Amir M. Rahmani, Seppo Virtanen, Hannu Tenhunen, Jouni Isoaho. " End-to-end security scheme for mobility enabled healthcare Internet of Things". *Future Generation Computer Systems* 64 (2016) 108–124
- [33] Fagen Li, Yanan Han, Chunhua Jin. " Practical access control for sensor networks in the context of the Internet of Things". *Computer Communications* 89–90 (2016) 154–164
- [34] Vu Mai, Ibrahim Khalil. " Design and implementation of a secure cloud-based billing model for smart meters as an Internet of things using homomorphic cryptography". *Future Generation Computer Systems*
- [35] Sergio Gusmeroli, Salvatore Piccione, Domenico Rotondi, "A capability-based security approach to manage access control in the Internet of Things" *Mathematical and Computer Modelling* 58 (2013) 1189–1205.
- [36] Jinshu Su, Dan Cao, Baokang Zhao, Xiaofeng Wang, Ilsun You, " ePASS: An expressive attribute-based signature scheme with privacy and an unforgeability guarantee for the Internet of Things" *Future Generation Computer Systems* 33 (2014) 11–18.

- [37] L.González-Manzano, JoséM.deFuentes, SergioPastrana, PedroPeris-Lopez, Luis Hernández-Encinas,"PAgIoT-Privacy-preserving Aggregation protocol for Internet of Things" Journal of Network and Computer Applications 71(2016) 59–7
- [38] Shruti Jaiswal, Daya Gupta, "Security Requirements for Internet of Things(IoT)" Proceedings of International Conference on Communication and Networks, Advances in Intelligent Systems and Computing 508, DOI 10.1007/978-981-10-2750-5_44
- [39] Donald Firesmith, "Engineering Security Requirements" in Journal of Object Technology, vol. 2, no. 1, January-February 2003, pages 53-68.
- [40] Kakali Chatterjee, Daya Gupta, Ashok De "A framework for development of secure software" CSI Transactions on ICT, 1(1), 143-157
- [41] G. Kotonya, I. Sommerville , "Requirement Engineering with viewpoints", 1995
- [42] I. Sommerville, "Software Engineering". Seventh edition 2003. ISBN - 8129708671. Pearson Education
- [43] OWASP, Open Web Application Security Project. Retrieved from <https://www.owasp.org/>
- [44] B. Forouzan, "Cryptography & Network Security". New York, USA: McGraw-Hill
- [45] Rajeev Piyare, "Internet of Things: Ubiquitous Home Control and Monitoring system using Android based Smart Phone" International Journal of Internet of Things 2013, 2(1):pages 5-11
- [46] Security Engineering. Retrieved from wikipedis.org/wiki/Security_engineering